**HỌC VIỆN KỸ THUẬT MẬT MÃ**

KHOA CÔNG NGHỆ THÔNG TIN



**THỰC TẬP CƠ SỞ CHUYÊN NGÀNH**

XÂY DỰNG ỨNG DỤNG PHÁT HIỆN

PHẦM MỀM ĐỘC HẠI ANDROID

|  |  |  |
| --- | --- | --- |
| Giảng viên hướng dẫn: | **Ths. Thái Thị Thanh Vân** | |
| Sinh viên thực hiện: | *Trần Gia Lương* | *CT030433* |
| *Trương Quốc Quân* | *CT030440* |
| *Vũ Thị Thanh Vân* | *CT030358* |

**Hà Nội, 2021-2022**

**LỜI CẢM ƠN**

Trong thời gian làm đề tài thực tập cơ sở chuyên ngành, chúng em đã nhận được nhiều sự giúp đỡ, đóng góp nhiệt tình của thầy cô, gia đình và bạn bè. Đầu tiên, chúng em xin được gửi lời cảm ơn chân thành và sâu sắc tới cô **Ths. Thái Thị Thanh Vân**, người đã tận tình hướng dẫn, chỉ bảo chúng em trong suốt quá trình thực hiện đề tài thực tập cơ sở chuyên ngành, giúp chúng em có thêm kiến thức chuyên môn cũng như học hỏi được tinh thần trách nhiệm, thái độ làm việc nghiêm túc, hiệu quả từ cô.

Chúng em cũng xin chân thành cảm ơn các thầy cô giáo trong Học viện Kỹ thuật Mật Mã nói chung cùng các thầy cô trong khoa Công nghệ thông tin nói riêng đã tận tình giảng dạy, truyền đạt cho chúng em những kiến thức và kinh nghiệm quý báu trong suốt thời gian học tập, giúp chúng em có được cơ sở lý thuyết vững vàng cũng như sự quan tâm và tạo mọi điều kiện thuận lợi cho chúng em trong quá trình thực hiện đề tài thực tập cơ sở chuyên ngành. Cuối cùng, chúng em xin chân thành cảm ơn gia đình và bạn bè, đã luôn tạo điều kiện, quan tâm, giúp đỡ, động viên chúng em trong suốt quá trình học tập và hoàn thành đề tài thực tập cơ sở chuyên ngành.

Với điều kiện thời gian cũng như kinh nghiệm còn hạn chế nên không thể tránh được những thiếu sót. Chúng em rất mong nhận được sự chỉ bảo, đóng góp ý kiến của các thầy cô để chúng em có điều kiện bổ sung, nâng cao ý thức của mình, phục vụ tốt hơn công tác thực tế sau này.

*Chúng em xin chân thành cảm ơn!*

**LỜI NÓI ĐẦU**

Vào cuối những năm 80, phần mềm chống mã độc (hay phần mềm chống virus) đã ra đời với sự ra đời của mã độc. Công nghệ được sử dụng trong giai đoạn mã độc này tương đối đơn giản, giúp máy tính dễ dàng phát hiện các mã độc đã biết trước. Hầu hết các phần mềm chống virus ban đầu đều áp dụng phương pháp so sánh tính năng và chỉ cần sử dụng chuỗi tính năng để hoàn tất phát hiện. Với sự phát triển của công nghệ mã độc, mã độc bắt đầu bị biến dạng trong quá trình lan truyền để tránh bị giết. Vào thời điểm này, số lượng biến thể của cùng một mã độc đã tăng lên đáng kể và đã thay đổi rất nhiều từ hình thức cấu trúc. Rất khó để phần mềm chống virus trích xuất một đoạn mã làm chữ ký cho mã độc.

Trong tình hình đó, chữ ký phổ rộng được ra đời, chữ ký phổ rộng chia chữ ký thành nhiều đoạn và sử dụng mã hóa cặp byte để phân chia các đoạn cần so sánh và các đoạn không cần so sánh. Tuy nhiên, cho dù đó là quét chữ ký hay chữ ký phổ rộng, cần phải trích xuất các tính năng sau khi lấy các mẫu mã độc, và sau đó mới có thể phát hiện. Điều này làm cho việc phát hiện mã độc có độ trễ nhất định và luôn đi theo mã độc. Để đối phó với các loại virus đã biết và cả chưa biết, quét heuristic đã ra đời. Quét heuristic sử dụng kinh nghiệm và kiến ​​thức đã có để tiến hành phát hiện mã nhị phân không xác định. Kỹ thuật này ghi lại mã độc có các tính năng độc hại mà các tệp nhị phân thông thường không có. Các hành vi, chẳng hạn như các tệp đọc và ghi khác thường, tự động kết thúc, v.v. Trọng tâm và khó khăn của quét heuristic là làm thế nào để trích xuất các đặc trưng hành vi độc hại của mã độc. Quét chữ ký, tìm kiếm các chữ ký phổ rộng và quét heuristic, cả 3 phương pháp tra diệt đều không có tệp nhị phân đang chạy thực tế, có thể được phân loại là phương pháp phát hiện tĩnh mã độc.

Với sự phát triển dần dần của công nghệ chống phần mềm độc hại và công nghệ điện toán đám mây đã được sử dụng ngày càng nhiều bởi các nhà cung cấp bảo mật, nhưng phương pháp phát hiện tĩnh mã độc vẫn là hiệu quả nhất đỡ tốn tài nguyên và thời gian, được sử dụng rộng dãi nhất để phát hiện mã độc.

Sau một thời gian thảo luận, nhóm sinh viên bọn em đã quyết định đăng kí chọn đề tài: “Xây dựng ứng dụng phát hiện phần mềm độc hại Android”

Bố cục đề tài được chia làm 3 chương như sau:

**Chương 1**: CƠ SỞ LÝ THUYẾT

**Chương 2**: MACHINE LEARNING

**Chương 3:** THỰC NGHIỆM HỆ THỐNG

KẾT QUẢ ĐẠT ĐƯỢC

**Sinh viên thực hiện:** Trần Gia Lương

Trương Quốc Quân

Vũ Thị Thanh Vân

**MỤC LỤC**

CHƯƠNG 1. CƠ SỞ LÝ THUYẾT 1

1.1. Tổng quan đề tài 1

1.1.1. Đặt vấn đề 1

1.1.2. Mục tiêu đề tài 2

1.2. Tổng quan về phần mềm độc hại 3

1.2.1. Mã độc 3

1.2.1.1. Khái niệm 3

1.2.1.2. Tác hại của phần mềm độc hại Android 3

1.2.2. Các dạng mã độc 3

1.2.2.1. Virus 3

1.2.2.2. Trojan horse 4

1.2.2.3. Ransomware- Mã độc tống tiền 6

1.2.2.4. Worm- Sâu máy tính 7

1.2.2.5. Rootkit 7

1.2.2.6. Botnet 7

1.2.2.7. Biến thể 8

1.2.3. Quan hệ giữa hai loại mã độc và các cơ chế hoạt động 8

1.2.4. Các cách phát hiện phần mềm độc hại trên Android 9

1.2.4.1. Phân tích tĩnh 9

1.2.4.2. Phân tích động 11

1.2.4.3. Phân tích lai 12

CHƯƠNG 2. MACHINE LEARNING VÀ MẠNG THÔNG TIN KHÔNG ĐỒNG NHẤT 14

CHƯƠNG 3. XÂY DỰNG ỨNG DỤNG PHÁT HIỆN PHẦM MỀM ĐỘC HẠI ANDROID 15

3.1. Xây dựng mô hình học máy 15

**DANH MỤC BẢNG**

**DANH MỤC HÌNH ẢNH**

[Hình 3.1: Xây dựng các ma trận cơ sở. 14](#_Toc91665565)

# CƠ SỞ LÝ THUYẾT

## Tổng quan đề tài

### Đặt vấn đề

Theo các số liệu thống kê mới nhất, Android hiện nay vẫn là hệ điều hành chiếm lĩnh phần lớn thị phần di động trên toàn cầu (87,7% - theo IDC 2018) và tỷ lệ này vẫn giữ nguyên cho đến năm 2021. Vì vậy, các thiết bị sử dụng hệ điều hành Android đã trở thành đối tượng bị tấn công của các ứng dụng độc hại android, gây ra mối đe dọa nghiêm trọng đối với việc rò rỉ dữ liệu cá nhân như vị trí người dùng, thông tin liên lạc, tài khoản, ảnh, vv... Thực trạng trên khiến việc xây dựng ứng dụng phát hiện phần mềm độc hại Android ngày càng trở nên cấp thiết.

Trước đây, hướng tiếp cận để giải quyết bài toán phát hiện phần mềm độc hại chủ yếu dựa vào kỹ thuật đối sánh mẫu truyền thống. Tuy nhiên, với sự phát triển nhanh chóng của các phương pháp học máy và trí tuệ nhân tạo, hướng nghiên cứu về việc phát triển các hệ thống tự động phát hiện phần mềm độc hại bằng cách sử dụng các kỹ thuật khai phá dữ liệu và học máy đang thu hút sự quan tâm của các nhà nghiên cứu trong lĩnh vực an ninh mạng. Vấn đề lớn nhất của bài toán phát hiện phần mềm độc hại sử dụng học máy đó là giải pháp phân tích, biểu diễn dữ liệu và trích chọn đặc trưng. Có hai cách tiếp cận để giải quyết vấn đề này, một là sử dụng kỹ thuật phân tích hành vi và hai là kỹ thuật phân tích chữ ký.

Kỹ thuật phân tích chữ ký dựa trên các dấu hiệu số (Digital Footprint) để đoán nhận mã độc. Tất cả các chương trình, dù là lành hay độc, đều có những dấu hiệu số đặc trưng của riêng mình. Các phần mềm diệt virus thường kết hợp với sử dụng một cơ sở dữ liệu lưu trữ các dấu hiệu số của các mã độc. Các phần mềm ấy sẽ đối sánh dấu hiệu số của file đang được quét với các dữ liệu trong cơ sở dữ liệu, Nếu tìm thấy dấu hiệu đó trong cơ sở dữ liệu, file đó sẽ được coi là độc. Khi một loại mã độc mới được phát hiện, dấu hiệu số của nó sẽ được thêm vào cơ sở dữ liệu của công ty viết ra phần mềm virus đó, và sẽ được chia sẻ cho cơ sở dữ liệu từ phía người dùng.

Hướng nghiên cứu phân tích hành vi dựa vào việc phân tích và đánh giá mã nguồn của ứng dụng được nghi ngờ. Trong bài toán đoán nhận mã độc Android, hướng nghiên cứu này dựa vào việc phân tích lời gọi API, quyền truy cập, lời gọi hệ thống. Các thông tin trên được biểu diễn bằng các mô hình mạng thông tin đồng nhất, nghĩa là nếu biểu diễn các dữ liệu này thành đồ thị thì các đỉnh trong đồ thị có chung một kiểu, và các cạnh của đồ thị đều biểu diễn một loại quan hệ. Trên mô hình mạng đã xây dựng, các nhóm sẽ đề xuất các thuật toán trích chọn đặc trưng phù hợp với mô hình học máy sử dụng. Hiệu quả của các hệ thống tự động phát hiện mã độc sử dụng mạng thông tin đồng nhất để biểu diễn dữ liệu và trích chọn đặc trưng cho kết quả dự đoán chính xác từ 85-92%. Nguyên nhân là do dữ liệu trong các hệ thống thực tế - đặc biệt là dữ liệu về mã độc thường phong phú, đa dạng và chứa đựng nhiều ý nghĩa tiềm ẩn khác. Nếu sử dụng mạng đồng nhất để biểu diễn thì có thể sẽ làm mất đi các thông tin ngữ nghĩa quan trọng trong mạng và việc trích chọn đặc trưng trên mô hình mạng đồng nhất đã không tận dụng tối đa các thông tin thu được để phát hiện mã độc, đặc biệt đối với các mã độc thế hệ mới. Vì vậy hướng nghiên cứu ứng dụng các kỹ thuật phân tích dữ liệu phức tạp để giải quyết thách thức trên đang được rất nhiều nhà nghiên cứu quan tâm. Một trong các công cụ thường được sử dụng để phân tích, biểu diễn dữ liệu phức tạp hiện nay là sử dụng mô hình mạng thông tin không đồng nhất (Heterogeneous Information Network- HIN).

Mạng thông tin không đồng nhất gồm nhiều loại đối tượng khác nhau và các liên kết (cạnh) trong mạng thông tin không đồng nhất có thể mang nhiều ý nghĩa khác nhau. Bởi vậy, việc khai phá các mạng thông tin không đồng nhất sẽ cho chúng ta biết thêm các tri thức còn tiềm ẩn trong các cấu trúc mạng, từ đó có thể ứng dụng trong nhiều lĩnh vực khác nhau. Cho đến hiện này, HIN đã được ứng dụng trong nhiều lĩnh vực khác nhau, như khai phá dữ liệu văn bản, khai phá dữ liệu sinh học, và đặc biệt ứng dụng trong bài toán phát hiện mã độc. Điển hình như nhóm nghiên cứu của [*Yanfang Ye*](https://www.researchgate.net/scientific-contributions/69831786_Yanfang_Ye) *và đồng nghiệp* đã sử mạng thông tin không đồng nhất để biểu diễn dữ liệu mã độc Android, mạng bao gồm 5 loại đỉnh (ứng dụng, API, IMEI, nhà sản xuất, chữ ký) và 5 loại cạnh (ứng dụng – API, ứng dụng-IMEI, IMEI-Nhà sản xuất, ứng dụng-nhà sản xuất, ứng dụng – chữ ký).

### Mục tiêu đề tài

Đối với môn học:

* Xây dựng được một hệ thống có thể phát hiện và phân loại các phần mềm độc hại
* Tìm hiểu phát hiện về mã độc

Đối với thành viên trong nhóm:

* Giúp sinh viên có khả năng làm quen với học máy
* Gia tăng kiến thức trong lĩnh vực bảo mật thông tin
* Cải thiện kỹ năng cần thiết như làm việc nhóm, giải quyết công việc

## Tổng quan về phần mềm độc hại

### Mã độc

#### Khái niệm

Mã độc hay còn gọi là phần mềm độc hại (malware/ malicious software) là một chương trình được bí mật chèn vào hệ thống mạng nhằm thực hiện các hành vi phá hoại. Khi xâm nhập thành công, mã độc có thể đánh cắp thông tin, làm gian đoạn hệ thống hoặc gây tổn hại tới tính bí mật, tính toàn vẹn và tính sẵn sàng của máy tính nạn nhân

Phần mềm độc hại là một dạng chương trình hoặc một đoạn chương trình tự cài đặt hoặc được cài đặt trên bất cứ thiết bị nào mà không cần sự cho phép của người dùng và thực hiện các chức năng mà người dùng không hề biết. Động cơ cơ bản của phần mềm độc hại là đánh cắp thông tin bảo mật từ điện thoại thông minh, khóa điện thoại thông minh, gửi tin nhắn SMS/MMS, thực hiện cuộc gọi đến đầu số cước phí đặc biệt, chia sẻ thông tin thông qua định vị GPS.

#### Tác hại của phần mềm độc hại Android

Tất cả những người dùng thiết bị di động chạy hệ điều hành Android có kết nối internet đều có thể trở thành nạn nhân của phần mềm độc hại. Phần mềm độc hại trên di động chủ yếu lấy cắp các dữ liệu nhạy cảm (như thẻ tín dụng, mật khẩu, nhật ký cuộc gọi, SMS) và theo dõi khách hàng. Ngoài ra, phần mềm độc hại có thể làm giảm tuổi thọ của pin hoặc giảm khả năng xử lý, chiếm quyền điều khiển trình duyệt, gửi tin nhắn trái phép và có thể vô hiệu hóa toàn bộ thiết bị. Một số trường hợp khách hàng mất tiền cước do phần mềm độc hại đăng ký dịch vụ game, thực hiện cuộc gọi quốc tế, …Tốc độ gia tăng nhanh chóng của các ứng dụng được phát triển trên nền tảng hệ điều hành này chính là môi trường thuận lợi cho các tin tặc phát triển các loại phần mềm độc hại mới với chiêu thức hoạt động tinh vi hơn nhằm trục lợi từ thông tin cá nhân người dùng hoặc thậm chí là tống tiền.

### Các dạng mã độc

Mã độc được chia thành nhiều loại tùy theo chức năng và cách thức lây nhiễm.

#### Virus

Virus chỉ là một dạng của mã độc nói chung. Điểm khác biệt nằm ở chỗ virus có khả năng lây lan cực nhanh, số lượng virus hiện nay chỉ chiếm chưa đến 10 % tổng số mã độc. Dưới đâu là 3 loại virus thường gặp

1. Virus Hoax

Cảnh báo giả về virus, cảnh báo giả này thường núp dưới dạng một yêu cầu khẩn cấp để bảo vệ hệ thống. Mục tiêu của cảnh báo virus giả là cố gắng lỗi kéo mọi người gửi cảnh báo càng nhiều càng tốt qua email. Cảng báo giả này không trực tiếp gây nguy hiểm. Tuy nhiên, các thư cảnh báo có thể chứa các chỉ dẫn về thiết lập lại hệ điều hành hoặc xóa file làm nguy hại tới hệ thống. Kiểu cảnh báo giả này cũng gây tốn thời gian và quấy rối bộ phận hỗ trỡ kỹ thuật khi có quá nhiều người gọi đến và yêu cầu dịch vụ.

1. Scripting Virus

Scripting virus là loại virus được viết bằng các ngôn ngữ script như VBScript, JavaScript, Batch script. Các loại virus này thường có đặc điểm dễ viết, dễ cài đặt. Chúng thường tự lây lan sang các file script khác, thay đổi nội dung cả các file html để thêm các thông tin quảng cáo, chèn banner… Đây cũng là một loại virus phát triển nhanh chóng nhờ sự phổ biến của Internet.

1. File Virus

Virus này thường lây vào các file thực thi (ví dụ file có phần mở rộng .com, .exe, .dll) một đoạn mã để khi file được thực thi, đoạn mã virus sẽ được kích hoạt trước và tiếp tục thực hiện các hành vi phá hoại, lây nhiễm. Loại virus này có đặc điểm lây lan nhanh và khó diệt hơn các loại virus khác do phải xử lý cắt bỏ, chỉnh sửa file bị nhiễm. Tuy nhiên, file virus có đặc điểm là chỉ lây vào một số định dạng file nhất định và phụ thuộc vào hệ điều hành.

#### Trojan horse

Tên của loại mã độc này được lấy theo một điển tích cổ. Trong cuộc chiến với người Tơ-roa, các chiến binh Hy Lạp sau nhiều ngày không thể chiếm được thành đã nghĩ ra một kế. Họ giả vờ giảng hòa rồi tặng người dân thành Tơ-roa một con ngựa gỗ khổng lồ. Sau khi ngựa gỗ được đưa vào thành, các chiến binh Hy Lạp từ trong ngựa gỗ chui ra đánh chiếm thành.

Đây cũng chính là cách mà các Trojan áp dụng: các đoạn mã của Trojan được “che giấu” trong các phần mềm máy tính thông thường để bí mật xâm nhập vào máy nạn nhân. Khi tới thời điểm thuận lợi, chúng sẽ đánh cắp thông tin cá nhân và chiếm quyền điều khiển máy tính… Bản chất của Trojan là không tự lây lan mà sử dụng phần mềm khác để phát tán.

Dựa vào cách hoạt động ta có thể phân chia Trojan thành 3 loại chính sau: BackDoor, Adware và Spyware.

1. Backdoor

Phần mềm Backdoor (cửa sau) là một dạng Trojan. Khi xâm nhập vào máy tính, backdoor sẽ mở ra một cổng dịch vụ cho phép tin tặc điều khiển máy tính nạn nhân. Tin tặc có thể cài phần mềm backdoor lên nhiều máy tính khác nhau thành một mạng lưới các máy bị điều khiển – Bot Net. Từ đó, thực hiện các vụ [tấn công từ chối dịch vụ DDoS](https://securitybox.vn/1353/12-loai-tan-cong-ddos-tan-cong-tu-choi-dich-vu-ddos/).

1. Adware- phần mềm quảng cáo

Đây là loại Trojan nhằm mục đích quảng cáo. Khi bị nhiễm adware, thiết bị có thể sẽ bị thay đổi trang chủ tìm kiếm, bị làm phiền bởi hàng loạt quảng cáo liên tục…

Adware là phần mềm được thiết kế để hiển thị quảng cáo lên màn hình thiết bị của người dùng, thường ngụy trang dưới dạng một chương trình hợp pháp hoặc ẩn mình trong một chương trình khác để lừa người dùng cài đặt nó. Khi xâm nhập thành công vào thiết bị, adware có thể thực hiện các tác vụ không mong muốn như: tự động mở tab mới, thay đổi trang chủ tìm kiếm, thu thập thông tin người dùng nhằm mục đích quảng cáo, chuyển hướng người dùng đến trang web khác.

1. Spyware- phần mềm gián điệp

Spyware (phần mềm gián điệp) là phần mềm dùng để đánh cắp thông tin của người dùng. Spyware thường được bí mật cài đặt trong các phần mềm miễn phí và phần mềm chia sẻ từ Internet. Một khi đã xâm nhập thành công, spyware sẽ điều khiển máy chủ và âm thầm chuyển dữ liệu người dùng đến một máy khác.

Spyware phần mềm độc hại xâm nhập vào thiết bị di động để ngấm ngầm thu thập thông tin bảo mật của người dùng và chuyển tiếp đến cho bên thứ ba. Spyware thường chạy ngầm trong hệ thống và âm thầm giám sát, thu thập thông tin nhằm phá hoại thiết bị di động cũng như quá trình truy cập Internet bình thường của người dùng.

#### Ransomware- Mã độc tống tiền

Graphical user interface, text

Description automatically generated

Hình: Thông báo tống tiền từ tin tặc.

Ransomware là một loại mã độc được dùng để ngăn chặn người dùng truy cập dữ liệu và sử dụng máy tính. Để lấy lại dữ liệu và quyền kiểm soát máy tính, nạn nhân cần chuyển tiền cho tin tặc. Đó là lý do ransomware còn được gọi là mã độc tống tiền. Ransomware thường xâm nhập qua email rác hoặc trang web lừa đảo. Trong một số trường hợp, ransomware được cài đặt cùng với Trojan để có thể kiểm soát nhiều hơn trên thiết bị của nạn nhân.

Sau khi đã xâm nhập vào máy tính của người dùng, ransomware sẽ rà soát máy tính và thực hiện một số tác vụ như:

* Mã hóa toàn bộ dữ liệu như dạng đuôi .doc, .xls, .pdf, file email.
* In một thông báo “tống tiền” ra ngoài màn hình chính hoặc để lại file có nội dung “tống tiền”.

#### Worm- Sâu máy tính

Worm là loại mã độc phát triển và lây lan mạnh mẽ nhất hiện nay. Worm thường được phát tán qua email. Các email này thường có nội dung giật gân và hấp dẫn để thu hút lượt click của người dùng. Nhờ những email giả mạo đó mà worm có thể lây lan theo cấp số nhân.

Nhận thấy khả năng lây lan mạnh mẽ của worm, những kẻ viết phần mềm độc hại này đã đưa thêm vào worm các tính năng khác như phá hoại hệ thống, ăn cắp thông tin… Có thể thấy, sức tàn phá của worm là vô cùng lớn

#### Rootkit

Rootkit lại được coi là một trong những loại mã độc nguy hiểm nhất. Rootkit là một chương trình máy tính được thiết kế để truy cập sâu vào hệ thống máy tính mà vẫn che giấu được sự hiện diện của nó và các phần mềm độc hại khác.

Nhờ rootkit, các phần mềm độc hại dường như trở nên “vô hình” trước những công cụ rà quét thông thường, thậm chí trước cả các phần mềm diệt virus. Việc phát hiện mã độc trở nên khó khăn hơn rất nhiều trước sự bảo vệ của rootkit.

#### Botnet

A picture containing diagram

Description automatically generated

Hình

Botnet là những máy tính bị nhiễm virus và bị điều khiển thông qua Trojan, virus… Tin tặc lợi dụng sức mạnh của những máy tính bị nhiễm virus để thực hiện các hành vi phá hoại và ăn cắp thông tin. Thiệt hại do Botnet gây ra thường vô cùng lớn.

Botnet có 2 loại là DNS và IRC

* DNS Bot: dễ thực hiện và điều khiển đơn giản không quá cầu kỳ, được dùng để chạy Bot trên nền Web. Nhưng vẫn bị hạn chế trong việc trao đổi thông tin giữa Bot master và các Bot
* IRC Bot: Giúp trao đổi thông tin giữa Bot Master và các Bot, điều khiển qua mạng chat IRC. Nhưng lại bị phụ thuộc vào các IRC và người quản trị Server

#### Biến thể

Một hình thức trong cơ chế hoạt động của virus là tạo ra các biến thể của chúng. Biến thể của virus là sự thay đổi mã nguồn nhằm tránh bị phần mềm diệt virus phát hiện hoặc làm thay đổi hành động của nó.

### Quan hệ giữa hai loại mã độc và các cơ chế hoạt động

Loại phần mềm độc hại thứ nhất và thứ hai của là một điều kiện tiên quyết cho các loại thứ ba, chúng đóng một vai trò cung cấp và duy trì cho các loại thứ ba trên hệ thống đích điều khiển từ xa một cách trái phép, việc thực hiện một mục đích cuối cùng của sự xâm nhập được thực hiện bởi một loại phần mềm độc hại thứ ba. Trong loại phần mềm độc hại đầu tiên, exploit, trojan, worm, bot và virus đều có thể bao gồm phần payload. Do đó, ngoài chức năng có được quyền kiểm soát từ xa đối với hệ thống đích, trojan, worm, bot và virus cũng có thể bao gồm loại phần mềm độc hại thứ hai và loại phần mềm độc hại thứ ba trong payload và phát hành chúng sau khi thực thi.

Một mối đe dọa hiện nay là có loại tấn công mang tên Blended Attck, sử dụng nhiều công nghệ liên quan đến phần mềm độc hại để tối đa hóa hiệu quả tấn công.

1. Trong giai đoạn đầu tiên để xâm nhập và giành được quyền điều khiển từ xa của hệ thống đích, các phương tiện xâm nhập như trojan, worm, bot và virus được kết hợp trong một chương trình thực thi cụ thể để có thể kết hợp giữa xâm nhập chủ động và xâm nhập thụ động, từ đó tăng cường đáng kể khả năng xâm nhập vào hệ thống đích.
2. Trong giai đoạn duy trì kiểm soát từ xa đối với hệ thống đích, việc sử dụng thành thạo các kỹ thuật liên quan đến rootkit cho phép rootkit và các phần mềm độc hại khác được bảo vệ để tồn tại trong hệ thống đích một cách đáng tin cậy và trong thời gian dài.
3. Khi hoàn thành mục đích xấu cụ thể, một loạt các chức năng phần mềm độc hại loại ba được tích hợp, để có thể thu thập thông tin, có thể gây thiệt hại cho hệ thống đích nếu muốn.

### Các cách phát hiện phần mềm độc hại trên Android

#### Phân tích tĩnh

Đối tượng chính của phân tích tĩnh trong phần mềm độc hại Android là kiểm tra các quyền, mã nguồn ứng dụng, bộ phận cấu thành, tài nguyên và mã nhận diện. Tất cả thông tin liên quan đến ứng dụng nằm trong tệp APK. Quyền truy cập, tài nguyên, mã, dịch vụ và tất cả thông tin khác được trích xuất từ tệp APK và được phân tích chính xác. Có nhiều công cụ có sẵn khác nhau cho phân tích tĩnh như apktool, aapt, dex2jar, jd-gui. Đối với việc áp dụng phân tích tĩnh vào ứng dụng Android, ta có thể tiếp cận theo hướng như sau: ứng dụng Android hầu như được xây dựng trên bốn thành phần bao gồm Activity biểu diễn các giao diện có thể tương tác với người dùng, Service thực thi các tác vụ ngầm, Content Provider truy cập cấu trúc dữ liệu cho ứng dụng và Broadcast Receiver xử lý các sự kiện hệ thống hoặc của người dùng. Các thành phần này tương tác với nhau thông qua các phương thức, ví dụ như startActivity () để kích hoạt giao tiếp liên thành phần (Inter-Component Communication – ICC). Phương thức của ICC nhận các tham số Intent chứa dữ liệu thành phần đích mà thành phần gốc muốn giao tiếp cùng. Có hai loại tương tác ICC là ICC rõ ràng, nơi Intent chứa tên thành phần đích và ICC không rõ ràng, nơi Intent chỉ chứa các mục tiêu tiềm năng, có khả năng xử lý, các Intent này cần được khai báo một Intent Filter trong tệp cấu hình AndroidManifest. Các phương pháp phân tích tĩnh có thể kể đến như sau:

* Phân tích tĩnh dựa trên chữ ký/mẫu: Tệp APK chứa chữ ký cụ thể. Mỗi tệp APK có chữ ký khác nhau. Chữ ký chứa thông điệp rút gọn của tệp APK. Nếu có bất kỳ sự thay đổi nào trong tệp APK thì thông điệp rút gọn của chữ ký cũng sẽ thay đổi và người ta có thể nhanh chóng phân tích được ứng dụng là độc hại. Chữ ký của phần mềm độc hại cũng có thể được thu thập để xác định phần mềm độc hại một cách nhanh chóng. Điểm hạn chế của phương pháp này chính là nó chỉ có thể phát hiện được các phần mềm độc hại hiện có, đã được biết tới; trong khi phần mềm độc hại luôn được phát triển và ngày càng tinh vi hơn.
* Phân tích tĩnh dựa trên tài nguyên: Tệp AndroidManifest.xml là một tệp tài nguyên chứa tất cả thông tin về các tài nguyên được sử dụng trong ứng dụng [15]. Tài nguyên của một ứng dụng chứa các module giao diện người dùng như widget, menu, layout, ...Tệp AndroidManifest.xml có trong tệp APK. Phần mềm độc hại chạy ngầm cần sự tương tác của người dùng từ các giao diện người dùng này. Vì vậy, giao diện người dùng cũng một phần quan trọng cần được phân tích chính xác.
* Phân tích tĩnh dựa trên thành phần: Ứng dụng Android được chia thành nhiều thành phần như nhà cung cấp nội dung, dịch vụ, đối tượng, hoạt động và thu nhận tín hiệu. Thông tin của tất cả các thành phần có sẵn trong tệp AndroidManifest.xml. Hầu hết phần mềm độc hại chạy dưới dạng dịch vụ ngầm và thu thập thông tin về đối tượng, hoạt động, đối tượng thu nhận. Vì vậy việc phân tích các thành phần này cũng rất quan trọng trong việc nhận ra các hành vi độc hại.
* Phân tích tĩnh dựa trên quyền truy cập: Android cung cấp mô hình dựa trên quyền truy cập cho việc triển khai bảo mật sẵn có. Tất cả các quyền truy cập được định nghĩa trong tệp AndroidManifest.xml. Các ứng dụng phải yêu cầu quyền truy cập như danh bạ, tin nhắn, internet, GPS, máy ảnh, ...Quyền truy cập đóng vai trò quan trọng trong bất kỳ ứng dụng Android nào. Một ứng dụng chỉnh ảnh đơn giản yêu cầu quyền ‘READ\_SMS’, sau đó thực hiện một số hành vi độc hại. Vì vậy, việc phân tích các quyền cũng rất quan trọng để nhận ra các hành vi độc hại.
* Kỹ thuật phân tích sử dụng công nghệ máy ảo: Kỹ thuật phân tích này kiểm tra hành vi ứng dụng, phân tích quyền kiểm soát và luồng dữ liệu để dò tìm những chức năng nguy hiểm. Nó có thể kiểm tra bytecode của ứng dụng và theo dõi các lời gọi API nhạy cảm. Tuy nhiên với kỹ thuật này, phân tích được thực hiện bởi các lệnh, nó cũng làm tốn pin và dung lượng lưu trữ hơn.

Phương pháp phân tích tĩnh trên Android đôi khi gặp phải những vấn đề cố hữu, mà việc khắc phục rất khó khăn và gần như bất khả thi, bao gồm:

* Mã biên dịch: Như đã biết, ứng dụng Android phần lớn được xây dựng từ ngôn ngữ Java nhưng được biên dịch thành Dex bytecode và thực thi trên máy ảo Dalvik hoặc ART, tuy nhiên các chương trình thu thập dữ liệu phục vụ phân tích tĩnh lại không có khả năng giải quyết Dex bytecode hoặc các định dạng hỗ trợ. Do đó, các chương trình này tuy hoạt động được với mã nguồn Java nhưng lại trở nên vô tác dụng trong hệ sinh thái Android. Ví dụ như phần mềm FindBug 5 được phát triển và hoạt động tốt trong việc phát hiện phần mềm độc hại Java, nhưng lại không thể đưa vào khai thác đối với Android.
* Điểm đầu vào: Không như các chương trình Java hay C, Android không có hàm main (), thay vào đó là các điểm đầu vào được gọi bởi Framework trong quá trình chạy. Đối lập với điều này, các ứng dụng phân tích phần mềm độc hại cần phải xác định điểm vào và sơ đồ hoá các liên kết. Với nhiều điểm đầu vào, việc xác định này rất khó khăn và không thể đảm bảo được sự liên kết giữa chúng.
* Vòng đời ứng dụng: Một vấn đề nữa đối với hệ điều hành Android đó chính là vòng đời của các thành phần ứng dụng. Mỗi một thành phần lại có một vòng đời riêng và đảm nhiệm một tác vụ hệ thống, và được gọi đến với nhiều kịch bản. Ví dụ, một ứng dụng khi không còn được sử dụng mà vẫn chiếm quá nhiều tài nguyên, hệ thống sẽ tự động đóng tiến trình mà không đi qua các hàm xử lý như vòng đời thông thường, các ứng dụng phân tích phần mềm độc hại sẽ gặp rất nhiều khó khăn khi xây dựng kịch bản hoạt động.
* Sự kiện: Sự kiện trên hệ điều hành Android có thể chia làm hai phần là sự kiện hệ thống (Cảnh báo pin, GPS …) và sự kiện người dùng (thao tác với UI). Các sự kiện này có thể xảy ra liên tục, chồng chéo nhau và có thể rơi vào bất kỳ thời điểm nào khi thiết bị hoạt động, khiến việc xây dựng và duy trì mô hình phân tích trở nên bất khả thi.
* Giao tiếp liên thành phần: Android có cơ chế giao tiếp giữa các thành phần, thậm chí còn diễn ra giữa hai ứng dụng khác nhau. Ứng dụng phân tích muốn quản lý các giao tiếp này yêu cầu một phương pháp phỏng đoán tiên tiến, nơi mà các liên kết được kích hoạt bằng các phương thức cụ thể với các tham số đa dạng. Mặt khác, các phương thức này còn được quản lý bởi vòng đời ứng dụng, hệ thống này chịu trách nhiệm trực tiếp trong quá trình khởi chạy.
* Thư viện: Thông thường, một ứng dụng Android sẽ sử dụng một hoặc nhiều thư viện để phục vụ người dùng, như thư viện quảng cáo hoặc mua bán. Những thư viện này có rất nhiều dòng mã, khiến kích thước của ứng dụng bị tăng lên đáng kể. Đối với phân tích tĩnh, việc đi sâu vào mã nguồn ứng dụng với các bộ thư viện này là không cần thiết và tiêu tốn rất nhiều thời gian, thậm chí còn nhiều hơn so với phân tích mã nguồn gốc. Mặt khác, phân tích các thư viện này tiềm ẩn nguy cơ thu thập các mẫu dương tính do cần yêu cầu các tài nguyên mà bị nhầm lẫn không cần thiết cho ứng dụng.

#### Phân tích động

Phân tích động bao gồm phân tích hành vi của ứng dụng. Nó kiểm tra sự tương tác của phần mềm độc hại với các tài nguyên di động và các dịch vụ như vị trí, mạng, gói, các hoạt động hệ điều hành. Đặc trưng của phân tích động bao gồm lời gọi hệ thống, lưu lượng mạng, luồng mạng, địa chỉ mạng. Phân tích động giám sát hành vi của hệ thống. Một số framework có sẵn để thực hiện phân tích động có thể kể đến như Ananas, TaintDroid, DroidScope, CopperDroid, Crowdroid. Các kỹ thuật phân tích động phổ biến được trình bày trong bảng sau:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Kỹ thuật phân tích | Cách thức hoạt động | Ưu điểm | Nhược điểm |
| 1 | Kỹ thuật phân tích dựa trên các hoạt động bất thường | Dựa trên việc quan sát hành vi thiết bị bằng cách theo dõi các tham số khác nhau và trạng thái của các thành phần thiết bị | Theo dõi các tham số khác nhau từ đó có cái nhìn rõ ràng về hệ thống | Số tham số tham gia càng nhiều thì các phép toán được yêu cầu càng nhiều |
| 2 | Kỹ thuật phân tích theo vết dữ liệu | Theo dõi nhiều nguồn dữ liệu nhạy cảm và phát hiện rò rỉ dữ liệu trong các ứng dụng di động | Theo dõi các dữ liệu nhạy cảm một cách hiệu quả | Không thực hiện theo dõi luồng kiểm soát |
| 3 | Kỹ thuật phân tích dựa trên mô phỏng | Phân tích ứng dụng dựa trên kỹ thuật giám sát thời gian chạy của máy ảo cấp hệ thống | Giám sát toàn bộ hệ thống bằng cách ở ngoài môi trường thực thi | Không phát hiện được các phần mềm độc hại mới |

1.2 Các kỹ thuật phân tích động

#### Phân tích lai

Phân tích lai là sự kết hợp của cả phân tích tĩnh và phân tích động. Quá trình bắt đầu với phân tích tĩnh. Phân tích tĩnh kiểm tra mã, quyền truy cập và các thành phần của ứng dụng. Sau đó, phân tích động thực hiện phân tích hành vi tổng thể của ứng dụng. Chỉ có một số framework dùng cho phân tích lai như Mobile Sandbox, Andrubis. Hệ thống AASandbox (Android Application Sandbox) đưa ra phân tích hai bước cho các ứng dụng Android. Một ứng dụng di động gửi đến AASandbox, tại đó nó thực hiện phân tích tĩnh và động trong chế độ ẩn (offline). Phân tích tĩnh vô hiệu hóa mã nhị phân ứng dụng cài đặt và sử dụng từng đoạn mã để tìm các mẫu nghi vấn. Phân tích động thực thi mã nhị phân trên bộ mô phỏng Android và ghi lại các lời gọi hệ thống.

Diagram

Description automatically generated

Hình 1.4 Kiến trúc của công cụ ANDRUBIS

Bên cạnh hai phương pháp phổ biến trên cũng có một số kỹ thuật khác cũng rất phổ biến như phân tích quyền ứng dụng, giám sát tuổi thọ pin hay gần đây nhất, với sự bùng nổ của công nghệ học máy và trí tuệ nhân tạo, việc phân tích đã được xây dựng tự động giúp rút ngắn tối đa thời gian cũng như chi phí phân tích và phát hiện phần mềm độc hại trên di động.

Phát hiện phần mềm độc hại là một yếu tố quyết định trong bảo mật hệ điều hành Android. Với sự phát triển nhanh chóng của các phương pháp học máy và trí tuệ nhân tạo, hướng nghiên cứu về việc phát triển các hệ thống tự động phát hiện phần mềm độc hại bằng cách sử dụng các kỹ thuật khai phá dữ liệu và học máy đang thu hút sự quan tâm của các nhà nghiên cứu trong lĩnh vực an ninh mạng. Với đề tài nghiên cứu này, chúng em sử dụng phương pháp phân tích tĩnh trong đó sử dụng thông tin về các lời gọi API và thông tin về mối quan hệ giữa các lời gọi API (các API cùng package, cùng block và cùng kiểu invoke), để xây dựng hệ thống phát hiện phần mềm độc hại dựa trên mạng thô

# MACHINE LEARNING VÀ MẠNG THÔNG TIN KHÔNG ĐỒNG NHẤT

# XÂY DỰNG ỨNG DỤNG PHÁT HIỆN PHẦM MỀM ĐỘC HẠI ANDROID

## Xây dựng mô hình học máy

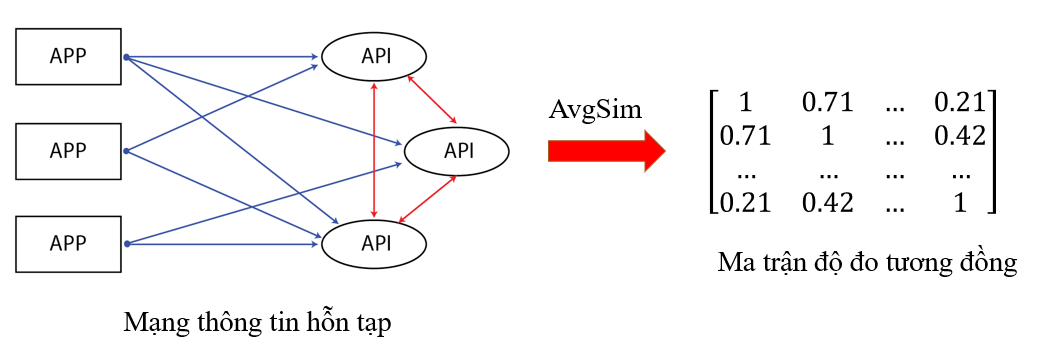
Do việc cơ sở dữ liệu của các họ mã độc còn hạn chế, nhóm sử dụng một tập dữ liệu chứa 589 ứng dụng android được chia thành sáu họ là: Benign, Adware, Botnet, Ransomware, Scareware và SMSmalware để xây dựng nên một mô hình học máy. Sau khi xây dựng lên được mô hình học máy này, nhóm sử dụng mô hình để phân chia 12000 ứng dụng android thành các họ riêng biệt để tạo nên một bộ dữ liệu học máy lớn hơn. Khi đã có bộ dữ liệu bao gồm 12000 ứng dụng đã được phân thia thành các họ, nhóm tiến hành xây dựng lại mô hình học máy dựa trên bộ dữ liệu 12000 ứng dụng này.

Vì vậy quá trình xây dựng mô hình học máy sẽ được chia thành hai giai đoạn bao gồm giai đoạn một: xây dựng mô hình semi – supervised learning để tạo cơ sở xây dựng một tập dữ liệu lớn hơn. Quy trình triển khai xây dựng mô hình được chia thành ba bước như sau:

* **Bước 1:** Thu thập dữ liệu và xây dựng các ma trận cần thiết. Nhóm đã sử dụng tập dữ liệu CICDataset và cụ thể tập dữ liệu có tên là CIC-AndMal2017, kết hợp cùng với một bộ android botnet để tạo nên bộ dữ liệu bao gồm 589 ứng dụng android, các ứng dụng android được chia thành sáu họ là: Benign, Adware, Botnet, Ransomware, Scareware và SMSmalware. Tiếp theo là sử dụng phầm mềm dịch ngược APK Tool để dịch ngược các ứng dụng android, thu được dạng mã smali và tiến hành trích xuất các API của từng ứng dụng. Từ dữ liệu đã được trích xuất, nhóm xây dựng nên các ma trận cần thiết bao gồm ma trận app – api, ma trận invoke, ma trận method và ma trận package

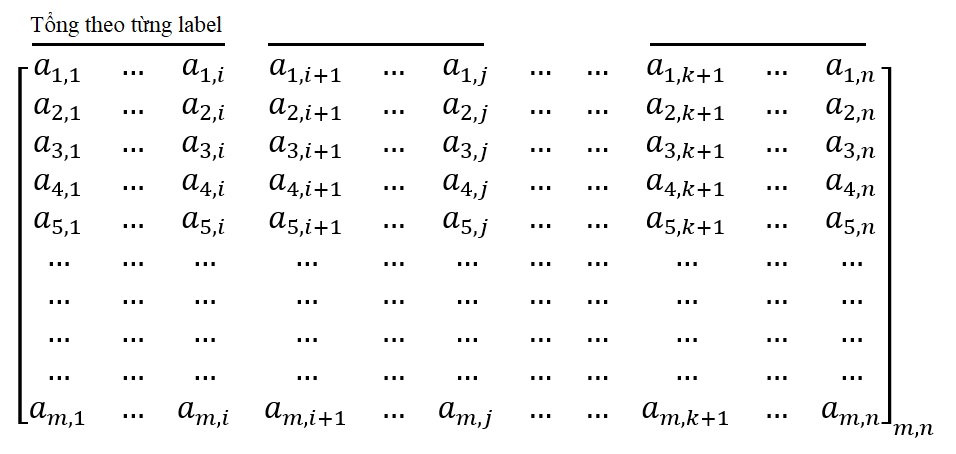
|  |  |
| --- | --- |
| Hình 3.1: Xây dựng các ma trận cơ sở. |  |

* **Bước 2:** Xây dựng mạng thông tin không đồng nhất và trích xuất vector đặc trưng. Từ bốn ma trận cơ bản, nhóm xây dựng mười sáu ma trận độ đo tương đồng và trích xuất vector đặc trưng dựa vào các ma trận độ đo trương đồng này.



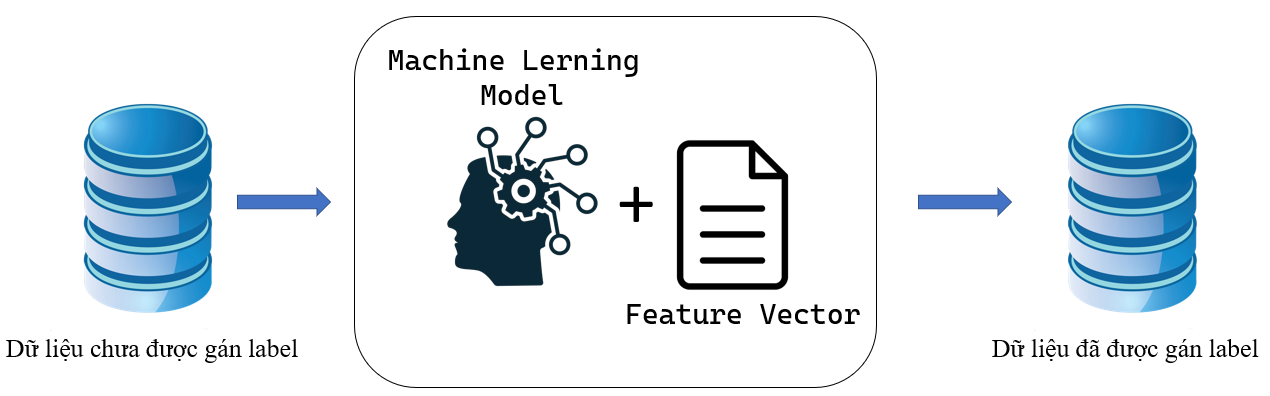
Hình 3.2: Xây dựng ma trận độ đo tương đồng

* **Bước 3:** Mỗi ma trận độ đo tương đồng sẽ cho ra sáu đặc trưng bằng cách gộp các hàng của ma trận đô đo tương đồng lại theo số lượng từng họ ứng dụng, cùng với mười sáu ma trận độ đo tương đồng kết hợp với sáu họ android ở tập dữ liệu sẽ tạo nên một ma trận vector đặc trưng sẽ có số chiều là chín mươi sáu. Ma trận vector đặc trưng này sẽ được sử dụng để phân loại 12000 ứng dụng android vào từng họ ứng dụng riêng biệt.



Hình 3.3: Mô tả cách tổng hợp ra các đặc trưng từ ma trận độ đo tương đồng

* **Bước 4:** Phân loại bộ 12000 ứng dụng andoid. Nhóm sử dụng mô hình vừa xây dựng được để phân loại 12000 ứng dụng android này thành 6 họ ứng dụng. với từng app android, nhóm sẽ đi trích xuất api để nhúng vào trong ma trận app – api, sau đó đưa qua AvgSim để thu được vector đặc trưng ứng với tập dữ liệu 589 ứng dụng android. Sau đó sẽ được mô hình học máy gán nhãn thành các họ ứng dụng riêng biệt. Kết thúc giai đoạn thứ nhất, ta thu được tập dữ liệu gồm 12000 ứng dụng đã được gán nhãn từng họ riêng biệt.



Hình 3.4: Gán label cho dữ liệu

Giai đoạn thứ hai giống hệt như giai đoạn một, chỉ khác ở chỗ là việc sử dụng tập dữ liệu lớn hơn, và vì khi dữ liệu ở trên mạng thông tin không đồng nhất càng nhiều, độ chích xác càng cao nên tỉ lệ chính xác tăng lên đáng kể. Cộng thêm việc nhóm đã có các ma trận app – api, invoke, method, package của tập dữ liệu 12000 ứng dụng này rồi nên không cần phải dịch ngược và trích xuất API nữa. Giai đoạn thứ hai bao gồm các bước như sau:

* Bước 1: Xây dựng ma trận độ đo tương đồng và vector đặc trưng. Từ bốn ma trận app – api, invoke, method, package. Nhóm sử dụng thuật toán để xây dựng lên mười sáu ma trận độ đo tương đồng, mỗi ma trận độ đo tương đồng sẽ cho ra sáu đặc trưng và kết quả thu được ma trận đặc trưng có số chiều là chín mươi sáu.
* Bước 2: Xây dựng hệ thống doán nhận có đầu vào là một app không biết trước, hệ thống sẽ dịch ngược ứng dụng và trích xuất lấy vector đặc trưng, sau đó đoán nhận và trả về kết quả.