7D-03

敵対的サンプルに対しても堅牢な制御フローグラフを用いた機械学習による Android マルウェア検知システム

美馬　隆志†　　篠埜　功‡

芝浦工業大学大学院　理工学研究科　電気電子情報工学専攻†‡

1.はじめに

マルウェアとはコンピュータに悪影響を及ぼすソフトウェアであり、様々な OS が対象となる。実行ファイルの形式や利用する API は OS 毎に異なる。そのため、検出時にファイルの中身を解析する場合は、OS ごとにマルウェアへの検出法が異なる。

AV-TEST[1]によると Android OS 対象のマルウェアが 2016 年頃から多く検出されており、Zhuo ら[2]は Android OS 対象のマルウェア（以下、Android マルウェア）を検出するシステムを提案した。このシステムは良性と悪性の APK ファイルの制御フローグラフを生成して、98.98%の精度で良性と悪性に分類する機械学習モデルを構築した。

この検出システムでは、入力として APK ファイルの制御フローグラフを利用するため、シグネチャ型の検出手法に対する攻撃者によるファイル内容の改変にも堅牢である。しかし、この機械学習モデルは敵対的サンプル(Adversarial Examples)に対しては脆弱である。敵対的サンプルとは、機械学習モデルに誤分類を引き起こさせるための改変を加えたデータのことである。仮に、攻撃者が既存の Android マルウェアから敵対的サンプルを作成した場合、この機械学習モデルは誤分類をする可能性がある。現在 Android マルウェアの敵対的サンプルはほとんど作成されていないと思われるが、将来 Android マルウェアの敵対的サンプルが増える可能性があり、この問題への対処法が求められる。

一般に、敵対的サンプルに堅牢な機械学習モデルを構築するための手法として、敵対的学習(Adversarial Training)が提案された。敵対的学習においては、敵対的サンプル無しで構築された機械学習モデルを、更に敵対的サンプルで学習する。

Alasmary らの研究[3]では、IoT マルウェアを対象として制御フローグラフの各ノードに対し、密度とレベルに基づいたラベル付けを行うことにより、敵対的サンプルを用いて 97.79%の精度で検出した。Alasmary らの調査[4]によると、Android マルウェアは IoT マルウェアと比較し、エッジ数が少なくノード数が多いという特徴があり、それぞれの制御フローグラフに適した検出手法が求められる。

そこで、本研究では Android OS を対象とし、悪性と良性の APK ファイルから実行可能な敵対的サンプルを作成し、機械学習モデルに追加で学習させることで、敵対的サンプルに対して堅牢な Android マルウェア検出システムを提案する。Zhuo らの手法[2]を第一筆者が実装した検出システムおよび本研究の提案システムで比較実験を行ったところ、Zhuo らの手法[2]では敵対的サンプルが一部入っているデータに対して正解率が 3 割ほど低下したのに対し、本研究の提案手法ではほとんど正解率が変化しなかった。

2.提案手法

1 節で述べた手法[2][3]などの制御フローグラフを用いた検出手法に効果的な敵対的サンプルの作成方法を提案する。また、この方法で作成される敵対的サンプルに対して堅牢な Android マルウェア検出システムを提案する。

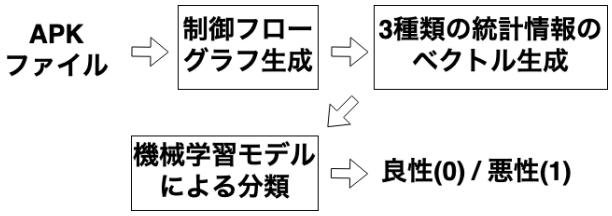

図 1:本研究で提案する Android マルウェア検出システムの処理の流れ

提案する検出システムを図 1 に示す。この検出システムは、APK ファイルから制御フローグラフを生成する段階、制御フローグラフから各種情報をベクトルとして抽出する段階、そして抽出したベクトルを用いて機械学習モデルで良性と悪性に分類する段階の 3 段階で構成される。

Android malware detection with a robust machine learning model against adversarial examples using control flow graphs
† Takashi Mima, Shibaura Institute of Technology
‡ Isao Sasano, Shibaura Institute of Technology

APK ファイルから制御フローグラフを生成する段階では、Arzt らが提案した FlowDroid の手法を利用する。制御フローグラフからライブラリ関数呼び出しの統計情報を表現した one-hot ベクトル、ライブラリ関数の呼び出し頻度を表現したベクトル、ライブラリ関数の呼び出し系列の一例を表現したベクトルの 3 種類のベクトルを生成する。最後に、アンサンブル学習した機械学習モデルにこれらの 3 種類の統計情報ベクトルを入力し、良性(0)と悪性(1)に分類する。

次に、制御フローグラフを用いた検出手法に効果的な敵対的サンプルは、Abusnaina らの GEA(Graph Embedding and Augmentation)[5]を参考に、新たに APK ファイルのエントリポイントを作成して、悪性と良性の APK ファイルのエントリポイントへ分岐させることで作成した。このように作成された敵対的サンプルを学習データに含めることで、制御フローグラフに特化した敵対的サンプルに対しても堅牢な機械学習モデルを構築する。

3. 評価実験

2 節で提案した Android マルウェア検出システムを実装し、このシステムの妥当性を確認するために実験を行った。この実装のソースコードおよび実験結果は https://github.com/task4233/detector_with_cfg で公開されている。実験データは Google Play から APK ファイルを 100 個収集し、また VirusShare.com[6]から 2,000 個の悪性 APK ファイルを収集した。ただし、制御フローグラフを抽出するための FlowDroid の制約により、Android API Level 22 以上を利用している APK ファイルを対象とした。上記のように収集した良性と悪性の APK ファイルからランダムに選択して結合し、敵対的サンプルを作成した。本実験の目的は、Zhuo らの手法[2]が敵対的サンプルに脆弱であることを示し、本研究の提案手法が敵対的サンプルに堅牢であることを示すことである。これらを示すために、Zhuo らの手法[2]で構築された機械学習モデルおよび本研究の提案手法で構築された機械学習モデルの精度評価を行う。精度評価は、既知のデータセットと敵対的サンプルでそれぞれ学習した機械学習モデルを、8:2 の Hold-out 法で検証し、精度、再現率、適合率、F 値を比較することで行う。

4. 実験結果と考察

3 節で述べた評価実験の結果を表 1 に示す。なお、数値は小数点第 4 位を四捨五入した値で、前から順に正解率、再現率、適合率、F 値である。

表 1 :Zhuo らの手法[2]と提案手法における正解率、再現率、適合率、F 値の比較

	敵対的サンプルを含まないデータセット	敵対的サンプルを含むデータセット
Zhuo らの手法[2]	0.976/0.978/0.998/0.988	0.675/1.000/0.675/0.805
提案手法	0.971/0.978/0.992/0.985	0.994/1.000/0.994/0.997

この結果より、Zhuo らの手法[2]は敵対的サンプルを含むデータセットに対する正解率が敵対的サンプルを含まないデータセットに対する正解率よりも 3 割ほど低下しており、本実験で評価用に作成した敵対的サンプルに対して脆弱である。一方で、提案手法では正解率が大きく変動していない。したがって、提案手法における機械学習モデルは 2 節で述べた方法で作成した敵対的サンプルに対して堅牢である。

5. まとめ

本研究では Android OS を対象とし、悪性と良性の APK ファイルから実行可能な敵対的サンプルを作成することにより、敵対的サンプルに対して堅牢な Android マルウェア検出システムを提案し実装した。また、提案手法の有効性を示すための評価実験を行い、Zhuo らの手法[2]は生成した敵対的サンプルに対して正解率が 3 割程度低下したが、提案手法は正解率がほとんど変化しないことを確認した。

参考文献

[1] “Malware Statistics & Trends Report”, https://www.av-test.org/en/statistics/, *AV-TEST*, 2023.
[2] Zhuo et al., “A Combination Method for Android Malware Detection Based on Control Flow Graphs and Machine Learning Algorithms”, *IEEE Access*, 7, 2019.
[3] Alasmary et al., “Soteria: Detecting Adversarial Examples in Control Flow Graph-based Malware Classifiers”, *IEEE ICDCS 2020*.
[4] Alasmary et al., “Analyzing and Detecting Emerging Internet of Things Malware: A Graph-Based Approach”, *IEEE IoT-J*, 6(5), 2020.
[5] Abusnaina et al, “Adversarial Learning Attacks on Graph-based IoT Malware Detection Systems”, *IEEE ICDCS 2019*.
[6] VirusShare.com., https://virusshare.com, 2023.