Attacchi Adversarial Machine Learning contro Sistemi di Malware Detection in Android

Nicolò Vescera

ANDROID OS

Android è il sistema operativo per dispositivi mobili più utilizzato al mondo:

- Dal 2011 per Smartphone
- Dal 2013 per Tablet
- Nel 2018 84,8% quota di mercato

DEXCODE VS SMALICODE

Java o Kotlin

Sono i linguaggi di **programmazione** con cui vengono scritte le App per Android.

```
int x = 42;
```

DexCode

È uno **speciale bytecode** per la JVM di Android.

```
13 00 2A 00
```

SmaliCode

Il dexcode può essere convertito in SmaliCode per essere più **leggibile**.

```
const/16 v0, 42
```

DEFINIZIONE MALWARE DETECTION

Un Malware Detector è un software che cataloga i programmi in **maligni** o **benigni** in base alla tipologia del loro codice.

Può essere visto come la seguente funzione:

$$D(p) = \begin{cases} maligno, & \text{se } p \text{ contiene codice malevolo} \\ benigno, & \text{se } p \text{ non contiene codice malevolo} \\ indeciso, & \text{se } D \text{ non riesce a classificare } p \end{cases}$$

TECNICHE DI MALWARE DETECTION

Le più comuni tecniche adottate nel problema della Malware Detection sono:

- Signature-based
- Anomaly-based
- Heuristic-based

ANDROID MALWARE DETECTION SYSTEMS

- Prendono in **input** l'applicazione **.apk** da valutare
- Estrazione del **feature vector**
- Classificazione

I Malware Detector che rappresentano lo stato dell'arte sono:

- Drebin
- Stormdroid
- MaMaDroid

ADVERSARIAL MACHINE LEARNING

Dato un classificatore ***M***, un record ***C*** di classe *y*:

$$M(A) \neq y$$
$$t.c.\ A = C + \delta,\ M(C) = y$$

METODOLOGIE DI ATTACCO

Poisoning

Questo attacco va a violare la fase di **training** aggiungendo **adversarial input** nel dataset.

Evasion

L'attaccante **modifica** un **input** con lo scopo di indurre il modello a predire una classe errata.

Model Extraction

Viola la **confidentiality** del modello riuscendo a **clonarlo** e recuperare informazioni sensibili tramite un numero finito di input.

ATTACCHI PRESENTI IN LETTERATURA

- *Adversarial Attacks on Mobile Malware Detection*

  Maryam Shahpasand et al. - 2019

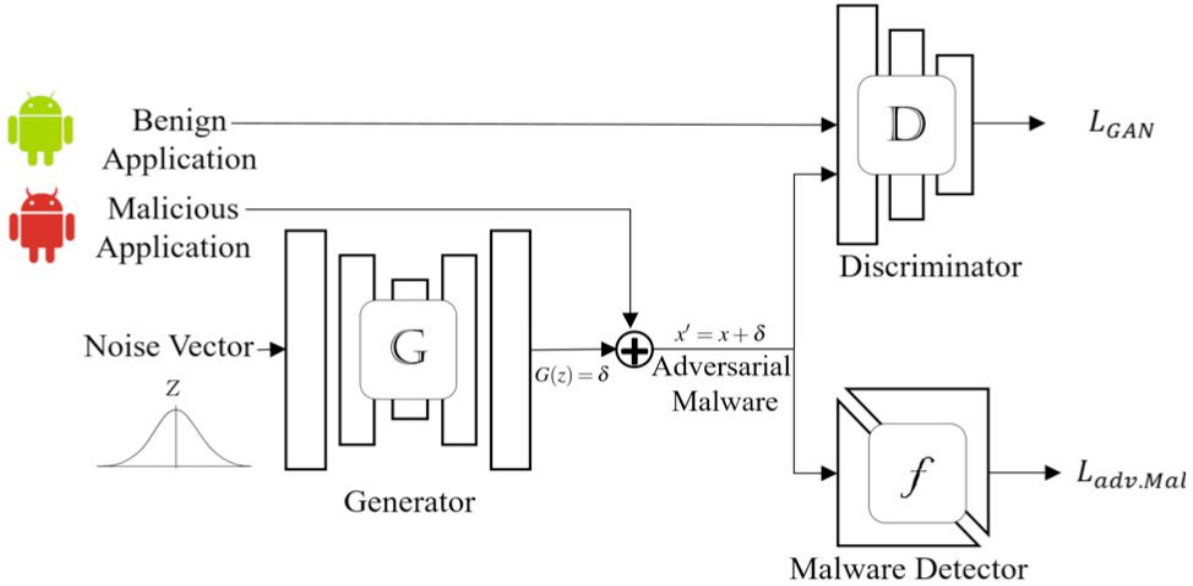
- *Android HIV: A Study of Repackaging Malware for Evading Machine-Learning Detection*

  Xiao Chen et al. - 2018

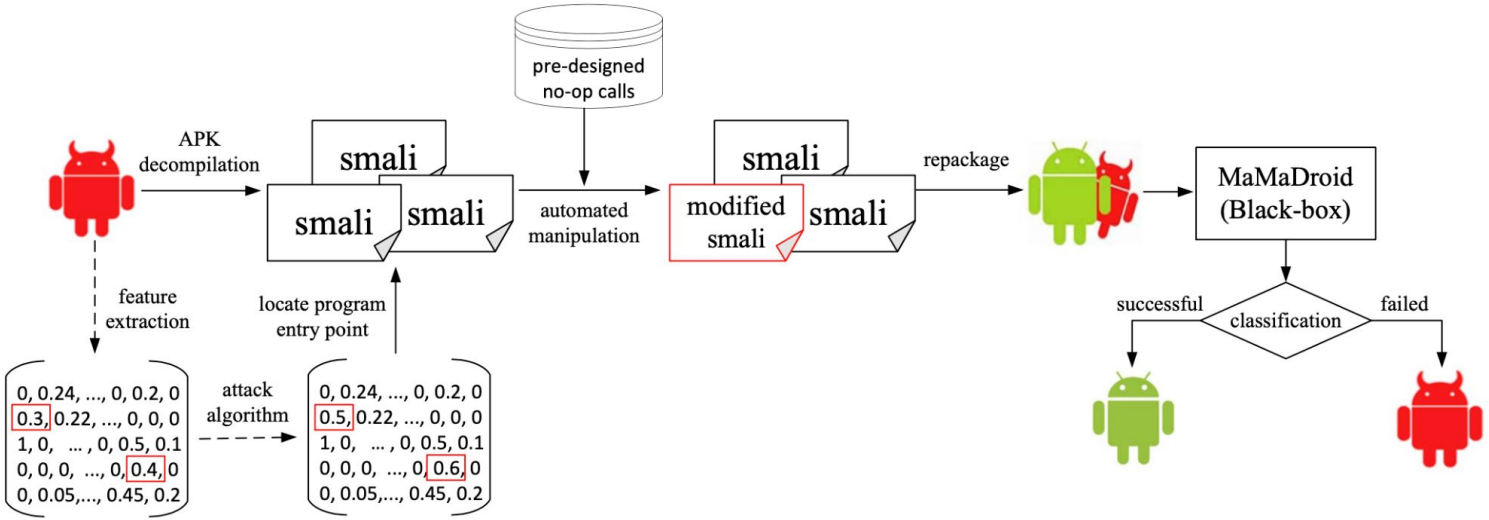
- *Adversarial Samples on Android Malware Detection Systems for IoT Systems*

  Xiaolei Liu et al. - 2019

ARCHITETTURA - SHAHPSAND ET AL.



MAMADROID ARCHITETTURA - XIAO CHEN ET AL.



PERTURBAZIONE - XIAO CHEN ET AL.

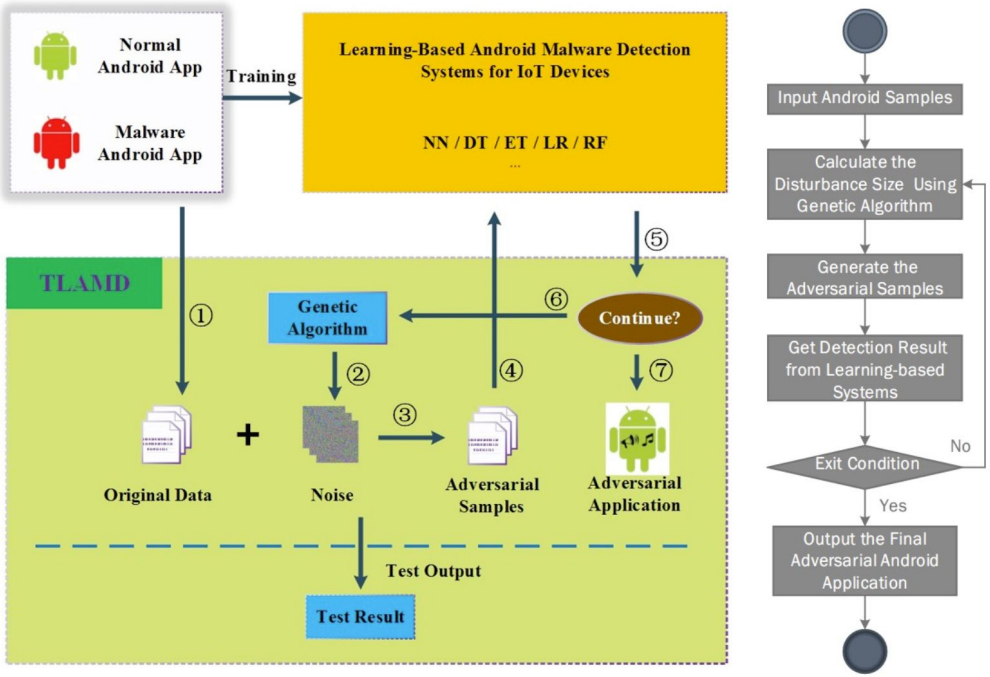
Java

```java
package android.os.mypack

public class Myclass {
    public static void callee() {}
    public static void caller() {
          callee();
          callee();
    }
}
```

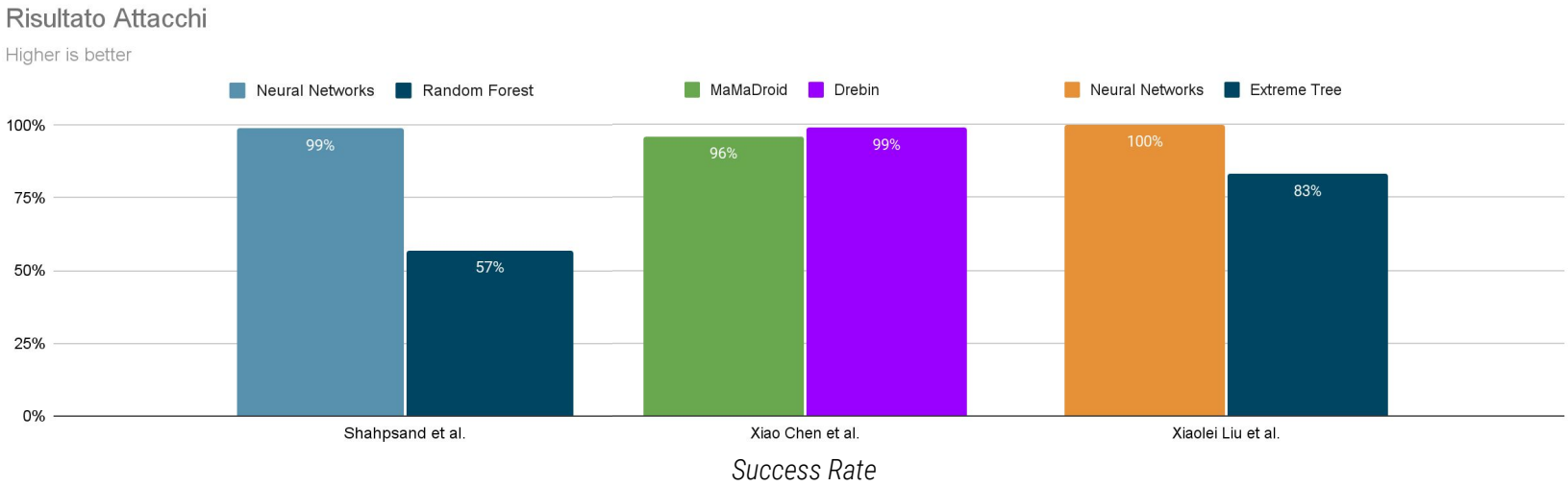
Smali

```
.class public
Landroid/os/mypack/Myclass;
.source "Myclass.java"

.method public static callee()V
    .locals 0
    return-void .end method

.method public static caller()V
    .locals 0
    .line 6 invoke-static {},

Landroid/os/mypack/Myclass;->callee()V
    return-void
.end method
```

ARCHITETTURA - XIAOLEI LIU ET AL.



RISULTATI


Risultato Attacchi
Higher is better
Neural Networks
Random Forest
MaMaDroid
Drebin
Neural Networks
Extreme Tree
100%
75%
50%
25%
0%
99%
57%
96%
99%
100%
83%
Shahpsand et al.
Xiao Chen et al.
Xiaolei Liu et al.
Success Rate

CONCLUSIONI

- Analizzato attacchi a Malware Detection System
- Malware Detector risultano estremamente vulnerabili
- Non possono essere considerati "sicuri"
- Studi futuri riguarderanno lo sviluppo di contromisure

GRAZIE PER L'ATTENZIONE!

Nicolò Vescera