

Machine Learning for Malware Analysis

Machine Learning
A.Y. 2018-2019



Dott. Luca Massarelli
massarelli@diag.uniroma1.it

CIS SAPIENZA
RESEARCH CENTER FOR CYBER INTELLIGENCE
AND INFORMATION SECURITY

Outline

- **Introduction**
- **Background**
- **Analysis objectives**
- **Applying machine learning to malware analysis**
- **Sample features for Android Malware**
- **The Drebin Dataset**
- **Homework**



CIS SAPIENZA
RESEARCH CENTER FOR CYBER INTELLIGENCE
AND INFORMATION SECURITY

What is a malware?



A malware is a malicious software that fulfills the deliberately harmful intent of an attacker

Nikola Milosevic. "History of malware". In: CoRR abs/1302.5392 (2013).

URL: <http://arxiv.org/abs/1302.5392>.



CIS SAPIENZA
RESEARCH CENTER FOR CYBER INTELLIGENCE
AND INFORMATION SECURITY

Malware typical characteristics

Often a malware:

- Is designed to damage users or systems;
- Exploits Software and Hardware Vulnerabilities;
- Uses Social Engineering to trick users;
- Can install other malware;
- Is controlled by a command and control server;



CIS SAPIENZA
RESEARCH CENTER FOR CYBER INTELLIGENCE
AND INFORMATION SECURITY

Different types

- Ransomware
- Dropper
- Trojan
- Advanced persistent threat
- ...



CIS SAPIENZA
RESEARCH CENTER FOR CYBER INTELLIGENCE
AND INFORMATION SECURITY

Beware of Malware!!!

*19% of all cyber attacks are
malware driven!*

(SERT Quarterly Threat Report Q2 2016)

*Globally, malware-based cyber
attacks grew of 85%*

*during the 1° semester 2017 with respect to the
2° semester 2016*

(CLUSIT Report 2017)



CIS SAPIENZA
RESEARCH CENTER FOR CYBER INTELLIGENCE
AND INFORMATION SECURITY

Mobile Malware

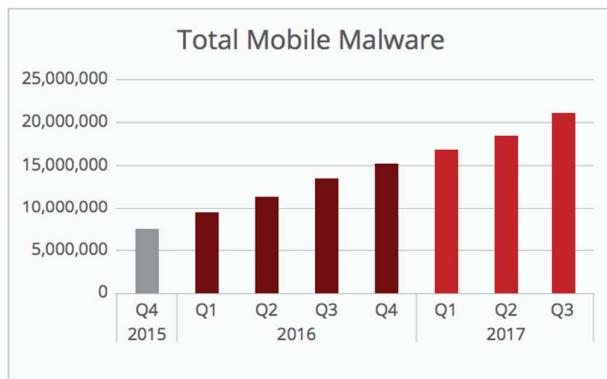


Figure 4. Total malware samples from 2015 – 2017.

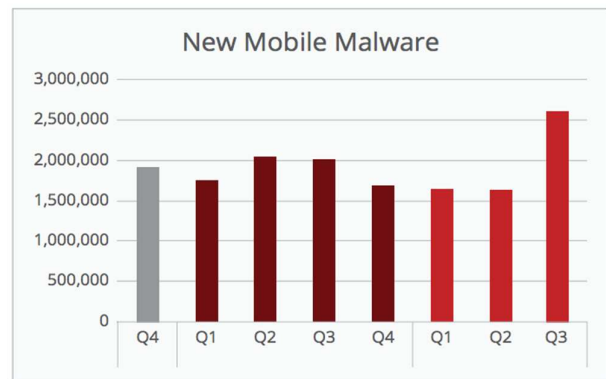


Figure 5. New malware samples from 2015 – 2017.

McAfee Mobile Threat Report Q1 18



CIS SAPIENZA
RESEARCH CENTER FOR CYBER INTELLIGENCE
AND INFORMATION SECURITY

Malware analysis and the role of Machine Learning

“Malware analysis concerns the study of malicious samples with the aim of developing a deeper understanding about several aspects of malware”

- Malware behavior
- How they evolve in time
- How they intrude target systems
- ...



CIS SAPIENZA
RESEARCH CENTER FOR CYBER INTELLIGENCE
AND INFORMATION SECURITY

Malware analysis and the role of Machine Learning

- Security defences are improving and evolving
- Nevertheless, malware are still succeeding

“Within the unceasing arm race between malware developers and analysts, each progress of security mechanisms is likely to be promptly followed by the realization of some evasion trick”



CIS SAPIENZA
RESEARCH CENTER FOR CYBER INTELLIGENCE
AND INFORMATION SECURITY

Malware analysis and the role of Machine Learning

“The easiness of overcoming novel defensive measures also depends on how well these measures capture malicious traits of samples”

Detection rules
based on pattern
matching

vs

Detection rules
to capture malicious
semantics



CIS SAPIENZA
RESEARCH CENTER FOR CYBER INTELLIGENCE
AND INFORMATION SECURITY

Malware analysis and the role of Machine Learning

- Defense-side goal:
*produce defensive technologies
as challenging as possible to overcome*
- Need to capture malicious aspects and traits
having the broadest scope
- **Machine Learning** is a natural choice to support
such a process of knowledge extraction



CIS SAPIENZA
RESEARCH CENTER FOR CYBER INTELLIGENCE
AND INFORMATION SECURITY

Malware analysis and the role of Machine Learning

- Plentiful availability of labelled samples
 - Very large training set
 - Key element to foster the adoption of machine learning for malware analysis
- Many works in literature have taken this direction, with a variety of approaches, objectives and obtained results...



CIS SAPIENZA
RESEARCH CENTER FOR CYBER INTELLIGENCE
AND INFORMATION SECURITY

Outline

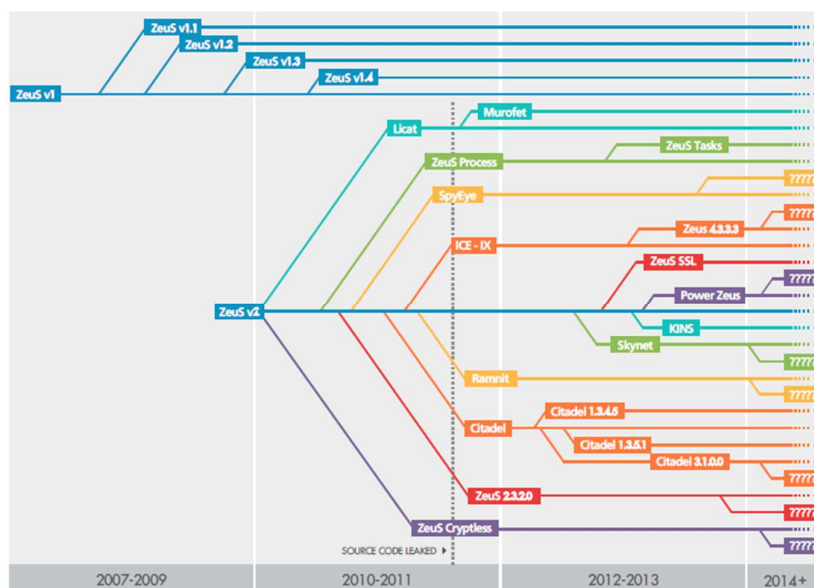
- Introduction
- **Background**
- Analysis objectives
- Applying machine learning to malware analysis
- Sample features for Android Malware
- The Drebin Dataset
- Homework



CIS SAPIENZA
RESEARCH CENTER FOR CYBER INTELLIGENCE
AND INFORMATION SECURITY

Malware variants

*Malware developers produce **variants** to minimize the effort required to evade updated security defences*



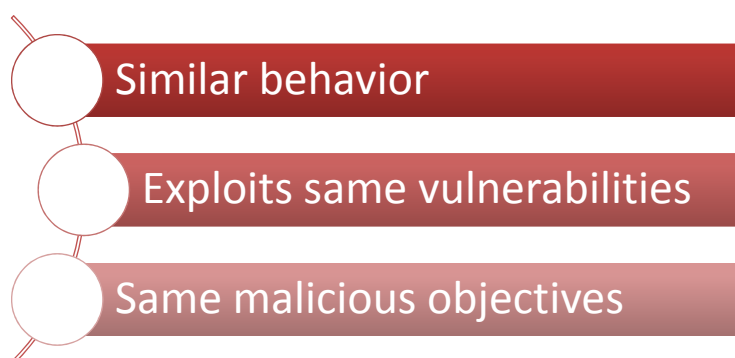
An “original”
malware evolves
in time through
the development
of variants (es:
Zeus)



CIS SAPIENZA
RESEARCH CENTER FOR CYBER INTELLIGENCE
AND INFORMATION SECURITY

Malware family

*The set of variants deriving from the same malware strain (i.e., “original” sample) is a malware **family***



CIS SAPIENZA
RESEARCH CENTER FOR CYBER INTELLIGENCE
AND INFORMATION SECURITY

Why to produce variants?

- Avoid-detection, changing a small part of the code.
- Re-use of source code, reduce the time required to produce a new malware.



CIS SAPIENZA
RESEARCH CENTER FOR CYBER INTELLIGENCE
AND INFORMATION SECURITY

Malware Family (Android example)



Package Name: com.requiem.slingshakLite
Activities: com.requiem...



Package Name: ca.rivalstudios.runboyrun
Activities: ca.rivalstudios.runboyrun...

However....



CIS SAPIENZA
RESEARCH CENTER FOR CYBER INTELLIGENCE
AND INFORMATION SECURITY

Malware Family (Android example)



Package Name: com.requiem.slingshakLite
Activities: com.requiem...
Services: com.GoldDream.zj.zjService
Receivers: com.GoldDream.zj.zjReceiver
Certificate:
61ed377e85d386a8df6e6b864bd85b0bfaa5af81
Relevant Strings:
<http://lebar.gicp.net/more.aspx?pid=9944&cid=1000>



Package Name: ca.rivalstudios.runboyrun
Activities: ca.rivalstudios.runboyrun...
Services: com.GoldDream.zj.zjService
Receivers: com.GoldDream.zj.zjReceiver
Certificate:
61ed377e85d386a8df6e6b864bd85b0bfaa5af81
Relevant Strings:
<http://lebar.gicp.net/more.aspx?pid=9944&cid=1000>



CIS SAPIENZA
RESEARCH CENTER FOR CYBER INTELLIGENCE
AND INFORMATION SECURITY

Obfuscation Techniques (Android example)

Obfuscation Techniques:

- Activity, Service and Receiver names can be changed and randomized;
- Applications can be signed with a different certificate;
- Binary code and application resources can be encrypted;



CIS SAPIENZA
RESEARCH CENTER FOR CYBER INTELLIGENCE
AND INFORMATION SECURITY

Not so simple...



Package Name: com.requiem.slingshakLite

Activities: com.requiem.se.1

Services: com.requiem.se.1

Receivers: com.requiem.se.1

Certificate:

94fg474u34d296n8pje9n060bi89n0brad5cf41

Relevant Strings:

EnCt2d5fcaad2bd889cb92be48ba0d67cc1e886= cf70fbd5fcaad2bd889cb92be48ba0X=GXQtvQ2gL



Package Name: ca.rivalstudios.runboyrun

Activities: ca.rivalstudios.runboyrun...

Services: com.rivalstudios.a.1

Receivers: com.rivalstudio.b.2

Certificate:

61ed377e85d386a8dfce6b864bd85b0bfaa5af81

Relevant Strings:

http://lebar.gicp.net/more.aspx?pid=9944&cid=1000



CIS SAPIENZA
RESEARCH CENTER FOR CYBER INTELLIGENCE
AND INFORMATION SECURITY

Signature-based analysis approaches

- Need to recognize already-known samples
 - If I know a sample is malicious, I want to detect its replicas
- Common techniques are signature-based
 - Hash of portions of code
 - Pattern matching on specific segments
 - Generally based on static characteristics
- Obviously malware can evade these techniques with obfuscation...



CIS SAPIENZA
RESEARCH CENTER FOR CYBER INTELLIGENCE
AND INFORMATION SECURITY

Machine learning analysis approaches

Machine learning permits to build malware analysis techniques that:

- Not need human support.
- Are resilient to obfuscation techniques.

In general machine learning permits to create malware analysis techniques based on the semantic of an application and not the code appearance.



CIS SAPIENZA
RESEARCH CENTER FOR CYBER INTELLIGENCE
AND INFORMATION SECURITY

Outline

- Introduction
- Background
- **Analysis objectives**
- Applying machine learning to malware analysis
- Sample features for Android Malware
- The Drebin Dataset
- Homework



CIS SAPIENZA
RESEARCH CENTER FOR CYBER INTELLIGENCE
AND INFORMATION SECURITY

Objectives of the analysis

- Malware detection is the most common objective
 - Respond to a real and urgent need: is this sample malicious? Should I worry?
 - Antiviruses are typical examples
- There can be situations where malware analysis is aimed at something different...



CIS SAPIENZA
RESEARCH CENTER FOR CYBER INTELLIGENCE
AND INFORMATION SECURITY

Malware variants detection

- A variant is *different enough* to evade detection, and *similar enough* to have the same behavior
- Detecting that a sample is a variant of a known malware is important
 - Removal and protection operations are likely the same
 - No need to investigate the sample further



CIS SAPIENZA
RESEARCH CENTER FOR CYBER INTELLIGENCE
AND INFORMATION SECURITY

Malware variants detection

- Variants selection
Given a malicious sample m , select from the available knowledge base the samples that are variants of m
- Families selection
Given a malicious sample m , select from the available knowledge base the families which m belongs to



CIS SAPIENZA
RESEARCH CENTER FOR CYBER INTELLIGENCE
AND INFORMATION SECURITY

Other objectives:

- Category Detection:
 - *Ransomware, trojan, ...*
- Novelty and similarity detection:
 - *What parts of the malware have been already seen?*
- Malware development detection:
 - *Discover ongoing developments of new malware;*
- Malware attribution:
 - *Who commissioned the development of a malware?*
- Malware triage:
 - *Prioritization of the analysis;*



CIS SAPIENZA
RESEARCH CENTER FOR CYBER INTELLIGENCE
AND INFORMATION SECURITY

Outline

- Introduction
- Background
- Analysis objectives
- **Applying machine learning to malware analysis**
- Sample features for Android Malware
- The Drebin Dataset
- Homework



CIS SAPIENZA
RESEARCH CENTER FOR CYBER INTELLIGENCE
AND INFORMATION SECURITY

Applying Machine Learning to malware analysis

- Supervised learning
 - Need for labelled training set
 - Relevant example: classify unknown samples in known malware families
- Unsupervised learning
 - No need for labelled training set
 - Relevant example: cluster samples to identify families



CIS SAPIENZA
RESEARCH CENTER FOR CYBER INTELLIGENCE
AND INFORMATION SECURITY

An example: Malware Detection

- Example: malware detection
 - Given a file, establish whether it is a malware
 - Two main types of analysis (hybrids are possible)
 - Static analysis
 - Dynamic analysis
 - It is a binary classification problem



CIS SAPIENZA
RESEARCH CENTER FOR CYBER INTELLIGENCE
AND INFORMATION SECURITY

An example: Malware Detection

- The goal is finding a function MD having
 - The set F of all possible files as domain
 - The set $\{P, N\}$ as codomain
 - Positive: the file is a malware
 - Negative: the file is not a malware

*Given a specific file type (subset of F),
how can we define MD?*



CIS SAPIENZA
RESEARCH CENTER FOR CYBER INTELLIGENCE
AND INFORMATION SECURITY

An example: Malware Detection

*Given a specific file type (subset of F),
how can we define MD?*

- Machine learning techniques provide means to find such a function
- Supervised learning allows to infer a function based on a labeled training dataset
- Given a function f to learn, with domain D and codomain C , the labelled training dataset (training set) is a set of pairs $\langle d, f(d) \rangle$, where $d \in D$ and $f(d) \in C$



CIS SAPIENZA
RESEARCH CENTER FOR CYBER INTELLIGENCE
AND INFORMATION SECURITY

An example: Malware Detection

In practice, supervised learning enables the learning of a function by providing a certain number of instances, each showing the expected output of the function given a specific input



CIS SAPIENZA
RESEARCH CENTER FOR CYBER INTELLIGENCE
AND INFORMATION SECURITY

An example: Malware Detection

- Several algorithms/models for supervised learning
 - Artificial neural networks
 - Decision trees, random forest
 - Support vector machines
 - Nearest neighbor
 - ...
- And several tools implementing them
 - Weka (www.cs.waikato.ac.nz/ml/weka)
 - Encog (www.heatonresearch.com/encog)
 - Sklearn
 - ...



CIS SAPIENZA
RESEARCH CENTER FOR CYBER INTELLIGENCE
AND INFORMATION SECURITY

An example: Malware Detection

- Instances of a domain can be complex
 - Android Application package
 - Huge execution trace of an application
 - Network traffic log of an application
- What is the actual input of the function to learn?
 - Each element can be represented by a fixed set of **features** (attributes) $\{a_1, \dots, a_n\}$ aimed at capturing all and only the characteristics that are relevant for the function to learn
 - Feature extraction is the process that, given an instance, returns its values for these features



CIS SAPIENZA
RESEARCH CENTER FOR CYBER INTELLIGENCE
AND INFORMATION SECURITY

An example: Malware Detection

- How to choose the set of features?
- What are the key characteristics for the function to learn?
- What are the specific cause-effect relationships that hold in that particular context?

This is where the intuition comes into play...



CIS SAPIENZA
RESEARCH CENTER FOR CYBER INTELLIGENCE
AND INFORMATION SECURITY

Evaluation Metrics

- Example: malware detection - accuracy metrics
 - Need to compare against some «ground truth»
 - Usually corresponds to the test set
 - For binary classification, there are four cases to be considered:

		Learned Function Output	
		Positive	Negative
Ground Truth	Positive	True Positive (TP)	False Negative (FN)
	Negative	False Positive (FP)	True Negative (TN)



Evaluation Metrics

- Example: malware detection - accuracy metrics
 - Meaning of true/false positive/negative for malware detection
 - True Positive
It is a malware, and I correctly detected it
 - False Positive
It is not a malware, but I thought it was
 - True Negative
It is not a malware, and I thought so too
 - False Negative
It is a malware, but I didn't detect it



Evaluation Metrics

- Example: malware detection - accuracy metrics
 - Precision: $TP / (TP + FP)$
 - How many files are real malware (TP) among those I considered as malware (TP + FP)?
 - «if I say it is a malware, then it really is a malware» (i.e., very few FP)
 - Recall: $TP / (TP + FN)$
 - How many malware did I spot (TP) among those in the test set (TP + FN)?
 - «if it is a malware, then I spot it» (i.e., very few FN)



CIS SAPIENZA
RESEARCH CENTER FOR CYBER INTELLIGENCE
AND INFORMATION SECURITY

Evaluation Metrics

- Example: malware detection - accuracy metrics
 - False Positive Rate: $FP / (FP + TN)$
 - How many files did I wrongly consider as malware (FP) among all the benign files (FP + TN)?
 - Accuracy: $(TP + TN) / (TP + FN + TN + FP)$
 - How many files did I classify correctly?
 - F-measure
 - $2 \cdot (precision \cdot recall) / (precision + recall)$
 - Can be interpreted as a weighted average of precision and recall
 - Best value: 1 worst value: 0



CIS SAPIENZA
RESEARCH CENTER FOR CYBER INTELLIGENCE
AND INFORMATION SECURITY

Outline

- Introduction
- Background
- Analysis objectives
- Applying machine learning to malware analysis
- **Sample features for Android Malware**
- The Drebin Dataset
- Homework



CIS SAPIENZA
RESEARCH CENTER FOR CYBER INTELLIGENCE
AND INFORMATION SECURITY

Sample features

The key element to apply successfully machine learning to malware analysis is a right choice of features.

- Features extraction depends on two factors:
 - Target operating system (Windows, Android, ...)
 - Type of analysis (Static, Dynamic or Hybrid)



CIS SAPIENZA
RESEARCH CENTER FOR CYBER INTELLIGENCE
AND INFORMATION SECURITY

Sample features for Android Malware

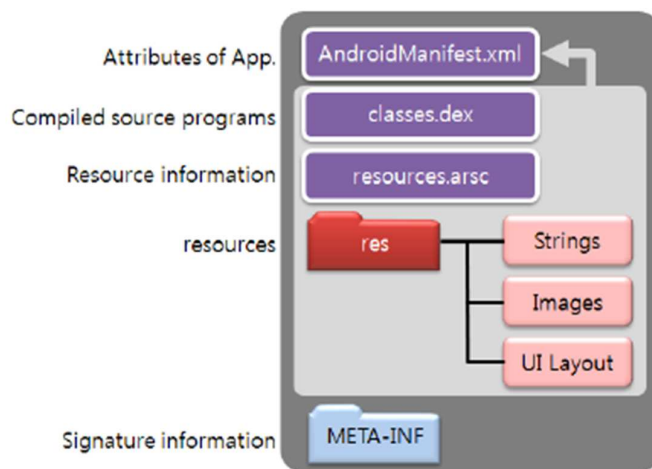
The structure of Android Operating system permits to extract some types of features that cannot be extracted in other environments.

*Mainly, the manifest.xml inside the package of an application (**apk**) contains very useful information...*



CIS SAPIENZA
RESEARCH CENTER FOR CYBER INTELLIGENCE
AND INFORMATION SECURITY

Android Application Fundamental



- All the components are important to detect a malware.
- A Malware can hide its malicious in different part of the applications, like resources or native code.



CIS SAPIENZA
RESEARCH CENTER FOR CYBER INTELLIGENCE
AND INFORMATION SECURITY

Static features:

- Features that can be extracted only by looking at the apk:
 - Components (*Activities, Services, Content Providers and broadcast receivers*);
 - Permissions;
 - API calls;
 - Strings;
 - Flow graph;



CIS SAPIENZA
RESEARCH CENTER FOR CYBER INTELLIGENCE
AND INFORMATION SECURITY

Dynamic features

- Features that can be extracted, executing the sample and looking at the execution traces:
 - Resource Consumptions;
 - System calls;
 - Download patterns;



CIS SAPIENZA
RESEARCH CENTER FOR CYBER INTELLIGENCE
AND INFORMATION SECURITY

Outline

- Introduction
- Background
- Analysis objectives
- Applying machine learning to malware analysis
- Sample features for Android Malware
- **The Drebin Dataset**
- Homework



CIS SAPIENZA
RESEARCH CENTER FOR CYBER INTELLIGENCE
AND INFORMATION SECURITY

Drebin

*DREBIN: Effective and Explainable Detection of
Android Malware in Your Pocket*

<https://www.tu-braunschweig.de/Medien-DB/sec/pubs/2014-ndss.pdf>

*Addressed the problem of malware family
classification and malware detection with a
support vector machine*



CIS SAPIENZA
RESEARCH CENTER FOR CYBER INTELLIGENCE
AND INFORMATION SECURITY

Drebin

- Used static analysis to extract features from samples;
- Used a dataset with 123,453 benign applications and 5,560 malware;
- The features extracted from each sample are public available;



CIS SAPIENZA
RESEARCH CENTER FOR CYBER INTELLIGENCE
AND INFORMATION SECURITY

Drebin – Extracted features

Features are extracted from the *manifest.xml* file and from the disassembled code.

Features can be divided in 8 different sets:

S_1, S_2, \dots, S_8



CIS SAPIENZA
RESEARCH CENTER FOR CYBER INTELLIGENCE
AND INFORMATION SECURITY

Drebin – Extracted features

- From *manifest.xml* file:
 - **S1: Requested hardware components** (GPS, camera, ...);
 - **S2: Requested Permissions** (Send sms, access to contacts,...);
 - **S3: App components** (Activities, Services, Content Providers, Broadcast receivers);
 - **S4 Filtered Intents** (Inter Process Communications handled by the sample e.g. *BOOT_COMPLETED*);



CIS SAPIENZA
RESEARCH CENTER FOR CYBER INTELLIGENCE
AND INFORMATION SECURITY

Drebin – Extracted features

- From the disassembled code:
 - **S5: Restricted API calls** (API calls whose access require a permission)
 - **S6: Used permissions** (Permissions effectively used by the application)
 - **S7: Suspicious API calls** (API calls who allow access to sensitive data e.g. *getDeviceId()*)
 - **S8: Network addresses** (Urls embedded in the code)



CIS SAPIENZA
RESEARCH CENTER FOR CYBER INTELLIGENCE
AND INFORMATION SECURITY

Drebin – Dataset organization

- The features extracted from each application of the dataset, can be downloaded as zip file from the course web-site.
- The zip file contains the features extracted from each application.
- Features extracted from an app are stored inside a text file whose name is the SHA1 Hash of the apk.



CIS SAPIENZA
RESEARCH CENTER FOR CYBER INTELLIGENCE
AND INFORMATION SECURITY

Drebin – Dataset organization



Each feature is composed by a prefix and a value, the prefix represent the set the feature belong to.

e.g.

Permission::android.permission.INTERNET

It belong to the set S2: required permission



CIS SAPIENZA
RESEARCH CENTER FOR CYBER INTELLIGENCE
AND INFORMATION SECURITY

Drebin – Dataset organization

Prefix	SET
feature	S1: Hardware components
permission	S2: Requested permission
activity service_receiver provider service	S3: App Components
intent	S4: Filtered Intents
api_call	S5: Restricted API calls
real_permission	S6: Used permission
call	S7: Suspicious API calls
url	S8: Network addresses



CIS SAPIENZA
RESEARCH CENTER FOR CYBER INTELLIGENCE
AND INFORMATION SECURITY

Drebin – Dataset organization

- The zip file contains also a dictionary file in csv format.
- In the dictionary there are the SHA1 Hash of the malware in the dataset and the family they belongs to.

eb1bcca87ab55bd0fe0cf1ec27753fddcd35b6030633da559eee42977279b8db, FakeInstaller
5010f34461e309ea1bc5539bb24fccc320576ce6d677a29604f5568c0a5e6315, Opfake
f1c8b34879b04dc94f0a13d33c1e1272bdf9141e56e19da62c1a1b27af128604, FakeInstaller
4e355df8f0843afc4a7bfc294ee4b1db9e9b896269c754c2d57dcb647dcd3efb, Opfake
ecf9c8520e13054bcc1b1a18cc335810f7eb97bdbe75fc204ad050228f805216, BaseBridge
255eae7859b0855b15de30e5405a2714837ac556c238bc009ac74c5bfa69714a, Nisev
54f2a636e000c55bb725d7e552a22117837c1676fb4b96dec135ae10e6f7049, BaseBridge

SHA1 HASH

FAMILY



CIS SAPIENZA
RESEARCH CENTER FOR CYBER INTELLIGENCE
AND INFORMATION SECURITY

Outline

- Introduction
- Background
- Analysis objectives
- Applying machine learning to malware analysis
- Sample features for Android Malware
- The Drebin Dataset
- Homework



CIS SAPIENZA
RESEARCH CENTER FOR CYBER INTELLIGENCE
AND INFORMATION SECURITY

Homework

- Use the Drebin dataset to address the problem of malware detection and malware classification.



CIS SAPIENZA
RESEARCH CENTER FOR CYBER INTELLIGENCE
AND INFORMATION SECURITY

Malware detection

- With all the applications in the dataset create a binary classifier whose output is
 - malware
 - non malware.
- Use the dictionary file as ground truth.
- Evaluate the performance of the classifier.



CIS SAPIENZA
RESEARCH CENTER FOR CYBER INTELLIGENCE
AND INFORMATION SECURITY

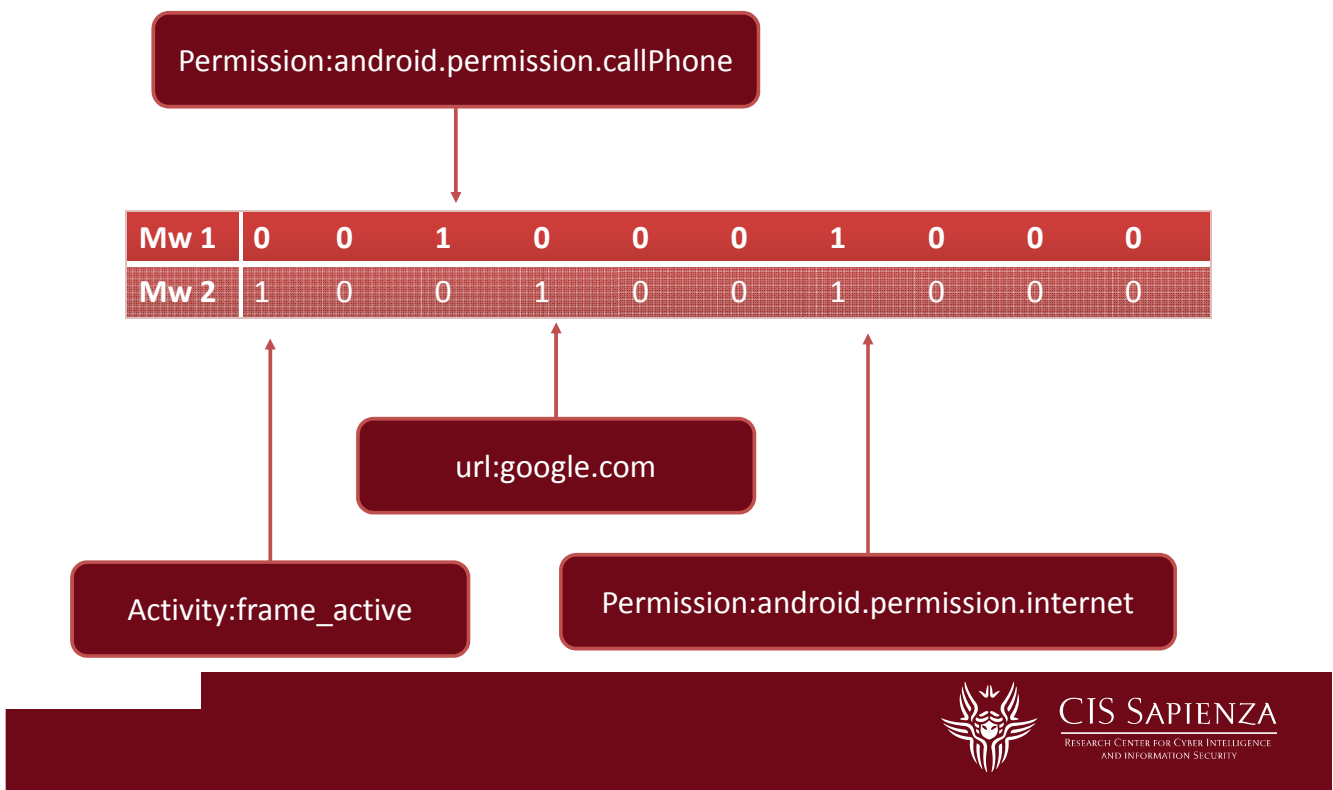
Malware detection - Hints

- Bayesian approach (e.g. SPAM filter), no need to match features with numerical value
- Other classification algorithm (SVM, Random Forest...), in this case you need to encode the features of each malware with a one-hot encoder.
- Try to use not all set of features but only a few (permissions, api calls, urls).
- Use the paper as reference.



CIS SAPIENZA
RESEARCH CENTER FOR CYBER INTELLIGENCE
AND INFORMATION SECURITY

One Hot encoder



Malware family classification

- Create a classifier that given in input the features of a malware output the family it belongs to.
- Use the dictionary as ground truth.
- Evaluate the performance of the classifier.



Malware family classification - Hints

- Select only malicious applications using the dictionary.
- Select only malware families that have more than 20 samples.
- Use a bayesian approach.
- Use different classification algorithm (SVM, Random Forest...)



CIS SAPIENZA
RESEARCH CENTER FOR CYBER INTELLIGENCE
AND INFORMATION SECURITY