Android malware is coming!

True but... what is malware, anyway?
Malware, All Malware: How Free Software Advocate Richard Stallman Sees Windows, Android And iOS

By Sumit Passary, Tech Times | May 27, 7:39 AM

Richard Stallman, a computer programmer and free software activist, brands famous operating systems such as iOS, Windows and Android as malware.

In an opinion piece in The Guardian, Stallman suggests that nearly all operating systems, whether desktop operating system or mobile operating system, can be considered malware. Stallman argues that any software that is not distributed free of cost is malware.

Stallman, who founded the Free Software Foundation, also made it clear in the opinion piece that he is not talking about any type of roughly 700,000

RSA 2015 Malware doesn't exist on Android, Google says, but Potentially Harmful Applications™ do.
Android and Google Play, they do indeed have great security measures... 
... but what if... malware (ok, PHA) DO NOT NEED TO BREAK THEM?
01 “Malicious code” in Android / Google Play

- Malware/adware that sends SMS premium messages.
- Steals information
- Makes you part of a botnet
- Click fraud
- Insane ads
- RATs
- ...

- All found outside and INSIDE Google Play. At least enough to keep producers creating them otherwise?
Antivirus is not exactly the technology we need for researching, discovering or analyzing new malware, frauds or threats.

Researchs work with “intelligence”. That is great, but...

...to build this intelligence, they are very “malware set dependant” (relying on VirusTotal or ContagioDump...), so they will be good detecting this kind of malware, but not the new one.

...regarding apps, traditionally, they have relied on permissions, number of permissions, code, urls.... and just that.

Some of them work with not so may apps to train this intelligence.

They feedback themselves with reputation, VT feedback, etc.

And of course... adware.
Adware is a tricky matter. Adware for Android may be very aggressive. Such aggressive that it could steal data, or flood you with popups since the telephone turns on.

**Antiviruses**

- Android developers use SDK to inject ads and have some revenues. They may configure this SDK so it is more or less aggressive.

- They usually mark this SDK as ADWARE. It does not matter how the developer have configured this ads to work into the app.

- They want to detect a lot, and get rid of ads for the user

**Google Play**

- Google Play IS OK WITH ADS, much more than Avs or even users. What may be wrong for an AV, is still ok with Google Play.

- Google Play wants to make money from this ads. But Google Play does not want to be “so OK that it bothers the user”...

- So, AFTER some complains, investigations, or any other criteria, it removes them.

So, basically, there is a grey area, and a conflict of interests.
01 What to do?

- Android apps are APK, which are sets of Java files packaged in a ZIP structure signed with a self signed certificate. We have identified and dissected most of the technical characteristics.

- Most of Android apps are hosted in Google Play, with a developer, comments, descriptions, images, versions, categories...

- There is plenty of information in three stages:
  
  Google Play -> The zip file itself -> cryptographic information

- **An app is not just an app but an app and its circumstances. Focus on WHO and HOW, aside the WHAT**

- Combining these three aspects we can get:
  
  A lot “checkpoints” of data, that may identify and classify an app.
  
  From these data, we can deduce the behavior of a developer.
  
  We collect all these data, make a database, and “shake it”.

This has to be seen as a complementary method to detect malicious behaviour and it is not intended to replace any existing ones.
How to do it? Are there any previous researches about this?

- We need to find discriminant “features” to distinguish between goodware and “malware” apps
- To be “really” fast, let’s consider features from the apps and its environment, that do no require code revision or installing the apps (and execution)
- Machine Learning using features (supervised learning) – SVM (Support Vector Machine)

Any difficulties?

Quantity:
- We need a huge dataset of apps to test different researches

Quality:
- Dataset Goodware and malware
- Features selection
02
Previous researches
Hello?… Is there anyone out there?
Droid Permission Miner: Mining Prominent Permissions for Android Malware Analysis

Aswini A. M.
Department of Computer Science & Engineering
SCMS School of Engineering & Technology
Karukutty, Ernakulam, India
aswinimohan95@gmail.com

Vinod P.
Department of Computer Science & Engineering
SCMS School of Engineering & Technology
Karukutty, Ernakulam, India
pvinod21@gmail.com

Jacques Klein
Yves Le Traon

approach DREBIN draws the user’s attention directly to rel-

ich
ors,

Panda, Sophos). We flag all applications as malicious that

\begin{table}[h]
\centering
\begin{tabular}{|c|c|c|}
\hline
          & DREB & \\
\hline
Full dataset & 93.90 & \\
Malgenome    & 93.90 & \\
\hline
\end{tabular}
\caption{Detection rates of DREBIN and anti-Virus scanners.}
\end{table}

\section{Datasets}
A total of 209 malicious samples are collected from the Contagiodump [2]. Likewise, benign applications were downloaded from various Internet sources (227 numbers).

are detected by at least two of the scanners. This proce-
rate of 1%, corresponding to one false alarm when installing

\begin{table}[h]
\centering
\begin{tabular}{|c|c|c|}
\hline
          & DREB & \\
\hline
Full dataset & 93.90 & \\
Malgenome    & 93.90 & \\
\hline
\end{tabular}
\caption{Detection rates of DREBIN and anti-Virus scanners.}
\end{table}

\section{Datasets}
A total of 209 malicious samples are collected from the Contagiodump [2]. Likewise, benign applications were downloaded from various Internet sources (227 numbers).

are detected by at least two of the scanners. This proce-
rate of 1%, corresponding to one false alarm when installing
Previous researches & results!

True but... what is malware, anyway?

These studies are all great. But, may we improve this in real life?

- They get a very good accuracy (up to 94%) and low False positives (down to 1%), detecting **specific malware**.
- They are very “malware set dependant” (relying on VirusTotal or ContagioDump...), so they will be good detecting this kind of malware.

- They have relied on **permissions, number of permissions and, in a way, code**.
- Some of them work **with not so many apps** to train.
But... What is malware?
Choosing malware set

VirusTotal is an excellent tool, but we think it needs to be “understood”... It is used for...

a) Comparing antivirus engines in a global or particular way. This is an awful idea. (Just read VirusTotal own help page...).

b) Cataloguing samples as malware. If it is “very detected” *(what does exactly that mean, anyway)* it is surely in the ”malware box”. Ok with it?
   • What is the right X? VT>1, VT>5, VT>10, VT>15? Engines are growing in VT...
   • If we use reputation of the engine as a factor... what is “reputation”? Vary famous?
   • Do we really take into account that some Avs simply do not work with Android?
   • Do we really penalize engines that detect “a lot”, so much that they may be false positives?

Setting as goodware samples with 0 detections: Depending of the freshness of the sample, samples are not detected by signatures.
02 What is malware?

Choosing the malware set

So far, for sure we just have "numbers" and subjective "quality" of engines... Did you know...

- VirusTotal sends the samples to all AV houses.
- From an AV standpoint, the first criteria for detecting such a flood of malware is precisely VirusTotal.
- Later, when they can remember rules, they may create a rule or analyze the sample.
- So, for some antivirus, VT is their first source, and... if researchers apply this criteria, this criteria is, in some way, feeding back itself.

Kaspersky defends false detection experiment

Claws in copy cat dust-up

By John Leyden, 10 Feb 2010  Follow 3,100 followers

Kaspersky Lab has defended its handling of a controversial experiment criticised by some as a marketing exercise of questionable technical value.

The Russian anti-virus firm created 20 innocent executable files, adding fake malware detections for ten of the sample, before uploading the files to online malware scanning service VirusTotal. VirusTotal routinely distributes samples of suspect files submitted to the service, which provides a useful tool for security pros to identify malware strains, to other vendors.

Ten days later, Kaspersky reports, 14 other vendors had added detection for the files it had deliberately (and falsely) labelled as malign.

Some representatives of security vendors, certainly those we spoke to at an event in Madrid last week, accused Kaspersky of attempting the suggest other vendors were copying its malware detections without applying anything like rigorous checks. In at least some cases, Kaspersky unveiled its findings to journalists attending its Moscow Labs rather than at a security conference or through peer review, further stoking negative sentiment about the experiment.

VirusTotal, which is run by Spanish penetration testing firm Hispasec: Sistemas, accused Kaspersky of misusing its services. Kaspersky's actions particularly irked VirusTotal because in the past it was accused of running a service that was misused by virus writers in order to test if their creations would escape detection.
What is malware?
Choosing the malware set

Is there a better way?

We do not know in PE world... but, let’s play in a smaller field... Android world, and Google Play town.

So, in this new field, some considerations should be taken into account.

So, what would be “malware” in Google Play?
How to build a good “malware set” so we can create a good classificator?
03

Signature accuracy and Gregariousness

A proposal to get better malware sets
Signature accuracy

Google Play knows its business, doesn’t it?

• Signature accuracy may be seen as another factor of quality assigned to a detection.

• It is based on using Google Play as a Judge, that, removing the app, validates in some way AVs detection. “Hey, you were right, this app should not be here, (although it will take a while for me to remove it”)

• “The more detected apps with a signature, that are eventually removed from Google Play, the more accurate the signature is”

• If these detected apps are not removed from Google Play, it means AVs were too aggressive, (it was just “tolerated adware”). This is why some of the studies say there is so much malware in Google Play.

• That is easy: NumberOfRetiredWithASignature/DetectedWithASignature = accuracy of the Signature.

    We could take “all the apps detected with high accuracy signatures”

    This would give us a nice malware set.
## Signature accuracy

Google Play knows its business, doesn’t it?

<table>
<thead>
<tr>
<th>AVEngine</th>
<th>Signature</th>
<th>nDead</th>
<th>detections</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>AegisLab</td>
<td>Android.Adware.Dowgin.D</td>
<td>1</td>
<td>1</td>
<td>1.0000</td>
</tr>
<tr>
<td>AntiVir</td>
<td>Android.Adware.Dowgin.X</td>
<td>1</td>
<td>1</td>
<td>1.0000</td>
</tr>
<tr>
<td>VIPRE</td>
<td>Android.Adware.Mulad.N</td>
<td>1</td>
<td>1</td>
<td>1.0000</td>
</tr>
<tr>
<td>ESET-NOD</td>
<td>Android.Riskware.SMSReg.AF</td>
<td>1</td>
<td>1</td>
<td>1.0000</td>
</tr>
<tr>
<td>VIPRE</td>
<td>Android.Riskware.SMSReg.CT</td>
<td>1</td>
<td>1</td>
<td>1.0000</td>
</tr>
<tr>
<td>ESET-NOD</td>
<td>Android.Riskware.SMSReg.W</td>
<td>1</td>
<td>1</td>
<td>1.0000</td>
</tr>
<tr>
<td>Comodo</td>
<td>Android.Trojan.Ewalls.A</td>
<td>1</td>
<td>1</td>
<td>1.0000</td>
</tr>
<tr>
<td>AVware</td>
<td>Android.Trojan.Mseg.A</td>
<td>1</td>
<td>1</td>
<td>1.0000</td>
</tr>
<tr>
<td>VIPRE</td>
<td>Trojan.Generic.KDV.690486</td>
<td>1</td>
<td>1</td>
<td>1.0000</td>
</tr>
<tr>
<td>AntiVir</td>
<td>Trojan.Script.603388</td>
<td>1</td>
<td>1</td>
<td>1.0000</td>
</tr>
<tr>
<td>VIPRE</td>
<td>BitCoinMiner</td>
<td>1</td>
<td>1</td>
<td>1.0000</td>
</tr>
<tr>
<td>Fortinet</td>
<td>Ganlet_1</td>
<td>1</td>
<td>1</td>
<td>1.0000</td>
</tr>
<tr>
<td>AegisLab</td>
<td>Kakay</td>
<td>1</td>
<td>1</td>
<td>1.0000</td>
</tr>
<tr>
<td>Sophos</td>
<td>SpyMob</td>
<td>1</td>
<td>1</td>
<td>1.0000</td>
</tr>
<tr>
<td>Agnitum</td>
<td>Trojan.Agent!0TdSTBuha+8</td>
<td>1</td>
<td>1</td>
<td>1.0000</td>
</tr>
<tr>
<td>AVware</td>
<td>Android-Adware/Gappusin</td>
<td>1</td>
<td>1</td>
<td>1.0000</td>
</tr>
<tr>
<td>Sophos</td>
<td>Android-Malicious/Eicar</td>
<td>1</td>
<td>1</td>
<td>1.0000</td>
</tr>
<tr>
<td>AhnLab-V3</td>
<td>Android-Malicious/SmsForw</td>
<td>1</td>
<td>1</td>
<td>1.0000</td>
</tr>
</tbody>
</table>

Source: elevenpaths.com
03 Signature accuracy

Google Play knows its business, doesn’t it?

• Yes, “accuracy” has some “problems”:

  • We have to improve it a bit, and introduce the concept of “Credibility” and “Participation”, so a “good” performance has been shown over time, detecting a minimum with success. How many times a signature should be used so it shows a good performance over time?

Credibility Based on Participation X Accuracy
### Signature accuracy /enhanced

<table>
<thead>
<tr>
<th>AVEngine</th>
<th>Signature</th>
<th>accuracy</th>
<th>nDetections</th>
<th>enhancedAccuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>MicroWorld-...</td>
<td>Android.Trojan.InfoStealer.DD</td>
<td>1.0000</td>
<td>5</td>
<td>0.6534395262100697</td>
</tr>
<tr>
<td>NANO-Antivi...</td>
<td>Trojan.Android.Funtasy.darpp</td>
<td>1.0000</td>
<td>5</td>
<td>0.6534395262100697</td>
</tr>
<tr>
<td>NANO-Antivi...</td>
<td>Trojan.Android.Leadbolt.dfxrl</td>
<td>1.0000</td>
<td>5</td>
<td>0.6534395262100697</td>
</tr>
<tr>
<td>Qihoo-360</td>
<td>Win32/Virus.DoS.44f</td>
<td>1.0000</td>
<td>5</td>
<td>0.6534395262100697</td>
</tr>
<tr>
<td>TrendMicro...</td>
<td>TROJ_GEN.F47V0626</td>
<td>1.0000</td>
<td>5</td>
<td>0.6534395262100697</td>
</tr>
<tr>
<td>Ad-Aware</td>
<td>Android.Adware.GingerMast...</td>
<td>0.8000</td>
<td>5</td>
<td>0.5227516209680557</td>
</tr>
<tr>
<td>AhnLab-V3</td>
<td>Android-Spyware/Flexion</td>
<td>0.8000</td>
<td>5</td>
<td>0.5227516209680557</td>
</tr>
<tr>
<td>AVG</td>
<td>Android/Deng.ES</td>
<td>0.8000</td>
<td>5</td>
<td>0.5227516209680557</td>
</tr>
<tr>
<td>BitDefender</td>
<td>Android.Adware.GingerMast...</td>
<td>0.8000</td>
<td>5</td>
<td>0.5227516209680557</td>
</tr>
<tr>
<td>Emsisoft</td>
<td>Android.Adware.GingerMast...</td>
<td>0.8000</td>
<td>5</td>
<td>0.5227516209680557</td>
</tr>
<tr>
<td>Fortin</td>
<td>ESET-NOD32 a variant of Android/Adware...</td>
<td>0.8000</td>
<td>5</td>
<td>0.5227516209680557</td>
</tr>
<tr>
<td>F-Prot</td>
<td>&lt;AndroidOS/Airpush.C</td>
<td>0.8000</td>
<td>5</td>
<td>0.5227516209680557</td>
</tr>
<tr>
<td>Jiang</td>
<td>F-Secure Android.Adware.GingerMast...</td>
<td>0.8000</td>
<td>5</td>
<td>0.5227516209680557</td>
</tr>
<tr>
<td>Soph</td>
<td>F-Secure Android.Adware.Minimob.A</td>
<td>0.8000</td>
<td>5</td>
<td>0.5227516209680557</td>
</tr>
<tr>
<td>ESET</td>
<td>F-Secure Android.Adware.Minimob.B</td>
<td>0.8000</td>
<td>5</td>
<td>0.5227516209680557</td>
</tr>
<tr>
<td>Soph</td>
<td>GData Android.Adware.GingerMast...</td>
<td>0.8000</td>
<td>5</td>
<td>0.5227516209680557</td>
</tr>
<tr>
<td>Soph</td>
<td>McAfee Artemis!40A755127451</td>
<td>0.8000</td>
<td>5</td>
<td>0.5227516209680557</td>
</tr>
<tr>
<td>ESET</td>
<td>MicroWorld... Android.Adware.GingerMast...</td>
<td>0.8000</td>
<td>5</td>
<td>0.5227516209680557</td>
</tr>
</tbody>
</table>
Gregariousness

Some Avs are more “reactive” than others

• What if the Avs detect a lot just when others do? (gregarious)
• What if Avs detect a lot of false positives? (eccentric)

We may improve this a little bit, trying to determine de “nature” of the signature. For the future, we need to penalize signatures that work a lot in “single” detections (eccentric) or just when any others detect (gregarious).

• So the optimum point is somewhere in the middle.
Gregariousness

Some Avs are more “reactive” than others.

What is the average number of engines detecting the same apps (in average) that this signature detects?

Given a signature:

• How many apps are detected by this signature?
• For this apps, how many others engines detect them (in average)?
• Basically: It is the average of the coefficient of detection.
### Gregariousness

Some AVs are more “reactive” than others

<table>
<thead>
<tr>
<th>AVEngine</th>
<th>Signature</th>
<th>nDetections</th>
<th>Gregariousness</th>
<th>Balanced</th>
</tr>
</thead>
<tbody>
<tr>
<td>Avira</td>
<td>Eicar-Test-Signature</td>
<td>2</td>
<td>0.875</td>
<td>0.362195</td>
</tr>
<tr>
<td>AegisL</td>
<td>EICAR (v)</td>
<td>2</td>
<td>0.875</td>
<td>0.362195</td>
</tr>
<tr>
<td>Cyren</td>
<td>EICAR_Test_File</td>
<td>2</td>
<td>0.875</td>
<td>0.362195</td>
</tr>
<tr>
<td>Sophos</td>
<td>Andr/Eicar-A</td>
<td>2</td>
<td>0.875</td>
<td>0.362195</td>
</tr>
<tr>
<td>AegisL</td>
<td>Test[not-a-virus]/Win32.EICAR</td>
<td>3</td>
<td>0.866667</td>
<td>0.385769</td>
</tr>
<tr>
<td>Ikarus</td>
<td>EICAR-ANTIVIRUS-TESTFILE</td>
<td>2</td>
<td>0.866667</td>
<td>0.385769</td>
</tr>
<tr>
<td>Ikarus</td>
<td>Testfile.AndroidOS.EICAR</td>
<td>1</td>
<td>0.866667</td>
<td>0.385769</td>
</tr>
<tr>
<td>AhnLab</td>
<td>EICAR-Test-File</td>
<td>3</td>
<td>0.866667</td>
<td>0.385769</td>
</tr>
<tr>
<td>AhnLab</td>
<td>EICAR тест file</td>
<td>3</td>
<td>0.866667</td>
<td>0.385769</td>
</tr>
<tr>
<td>Zillya</td>
<td>EICAR.TestFile</td>
<td>3</td>
<td>0.866667</td>
<td>0.385769</td>
</tr>
<tr>
<td>Zoner</td>
<td>EICAR.TestFile-NoVirus</td>
<td>3</td>
<td>0.866667</td>
<td>0.385769</td>
</tr>
<tr>
<td>AhnLab</td>
<td>EICAR-Test-File (not a virus)</td>
<td>4</td>
<td>0.85</td>
<td>0.432373</td>
</tr>
<tr>
<td>Agnitum</td>
<td>EICAR_test_file</td>
<td>4</td>
<td>0.85</td>
<td>0.432373</td>
</tr>
<tr>
<td>AhnLab-V3</td>
<td>Android-Malicious/Eicar</td>
<td>1</td>
<td>0.85</td>
<td>0.432373</td>
</tr>
<tr>
<td>AhnLab-V3</td>
<td>Android-Test/Eicar</td>
<td>3</td>
<td>0.85</td>
<td>0.432373</td>
</tr>
<tr>
<td>AntiVir</td>
<td>Avast EICAR Test-NOT virus!!!</td>
<td>4</td>
<td>0.85</td>
<td>0.432373</td>
</tr>
<tr>
<td>AntiVir</td>
<td>AVG EICAR_Test</td>
<td>4</td>
<td>0.85</td>
<td>0.432373</td>
</tr>
<tr>
<td>AntiVir</td>
<td>Baidu-International EICAR.Test.File</td>
<td>4</td>
<td>0.85</td>
<td>0.432373</td>
</tr>
<tr>
<td>AntiVir</td>
<td>BitDefender EICAR-Test-File (not a virus)</td>
<td>4</td>
<td>0.85</td>
<td>0.432373</td>
</tr>
<tr>
<td>AntiVir</td>
<td>CAT-QuickHeal EICAR.TestFile</td>
<td>4</td>
<td>0.85</td>
<td>0.432373</td>
</tr>
<tr>
<td>AntiVir</td>
<td>ClamAV Eicar-Test-Signature</td>
<td>4</td>
<td>0.85</td>
<td>0.432373</td>
</tr>
</tbody>
</table>

- If the apps are very detected, Gregariousness would go to 1.
- If the signature is unique, your Gregariousness would go to 0.
- Balanced goes to 1 when the signature is “balanced” (neither gregarious nor eccentric).
- Balanced goes to 0 when the signature is close to any of the edges of a Gauss function.
### 03 Gregariousness / Balanced

Some AVs are more “reactive” than others

<table>
<thead>
<tr>
<th>AVEngine</th>
<th>Signature</th>
<th>Balanced</th>
<th>nDetections</th>
<th>enhancedBalanced</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baidu-International</td>
<td>Trojan.Android.Plankton.I</td>
<td>0.906888</td>
<td>107</td>
<td>0.8764747302696492</td>
</tr>
<tr>
<td>Rising</td>
<td>DEX:PUF.Plankton!1.9DAE</td>
<td>0.893638</td>
<td>182</td>
<td>0.8757720740151775</td>
</tr>
<tr>
<td>F-Prot</td>
<td>AndroidOS/Apperhand.A</td>
<td>0.872509</td>
<td>230</td>
<td>0.8586480437113408</td>
</tr>
<tr>
<td>Fortinet</td>
<td>Android/Apperhand.AAtr</td>
<td>0.868443</td>
<td>303</td>
<td>0.8579308874490152</td>
</tr>
<tr>
<td>NANO-Antivirus</td>
<td>Trojan.Dex.Startapp.ctgcbx</td>
<td>0.886432</td>
<td>110</td>
<td>0.8574892964166416</td>
</tr>
<tr>
<td>AntiVir</td>
<td>Android/Plankton.C.Gen</td>
<td>0.86681</td>
<td>336</td>
<td>0.8573365193301186</td>
</tr>
<tr>
<td>AVG</td>
<td>Android/Plankton</td>
<td>0.862285</td>
<td>216</td>
<td>0.8477139621037821</td>
</tr>
<tr>
<td>Ikarus</td>
<td>AndroidOS.AdWare.Plankton</td>
<td>0.908439</td>
<td>46</td>
<td>0.8405925646328143</td>
</tr>
<tr>
<td>Bkav</td>
<td>Android.Adware.ApperhandAds</td>
<td>0.888487</td>
<td>65</td>
<td>0.8404792425384358</td>
</tr>
<tr>
<td>F-Secure</td>
<td>Application:Android/Counterclank</td>
<td>0.850839</td>
<td>228</td>
<td>0.8372056125400107</td>
</tr>
<tr>
<td>DrWeb</td>
<td>Adware.Startapp.6.origin</td>
<td>0.857509</td>
<td>149</td>
<td>0.8366607529987036</td>
</tr>
<tr>
<td>Sophos</td>
<td>Android Apperhand</td>
<td>0.844874</td>
<td>321</td>
<td>0.8352138454825138</td>
</tr>
<tr>
<td>Commtouch</td>
<td>AndroidOS/Plankton_A.gen!Eldorado</td>
<td>0.847963</td>
<td>202</td>
<td>0.832658662261494</td>
</tr>
<tr>
<td>Ikarus</td>
<td>AndroidOS.AdWare.Apperhand</td>
<td>0.832447</td>
<td>179</td>
<td>0.8155312068458045</td>
</tr>
<tr>
<td>ESET-NOD32</td>
<td>a variant of Android/Plankton.I</td>
<td>0.826355</td>
<td>268</td>
<td>0.8150636092454455</td>
</tr>
<tr>
<td>F-Prot</td>
<td>AndroidOS/Plankton.D</td>
<td>0.836413</td>
<td>85</td>
<td>0.8014071129206428</td>
</tr>
<tr>
<td>Tencent</td>
<td>Adware.Android.Apperhand.a</td>
<td>0.872193</td>
<td>41</td>
<td>0.7997691393094328</td>
</tr>
<tr>
<td>Ikarus</td>
<td>AdWare.AndroidOS.Apperhand</td>
<td>0.824106</td>
<td>98</td>
<td>0.7940238691219053</td>
</tr>
</tbody>
</table>
### All together now

<table>
<thead>
<tr>
<th>AV Engine</th>
<th>nDetections</th>
<th>enhancedAccuracy</th>
<th>enhancedBalanced</th>
<th>enhancedAxenEnhancedBalance</th>
</tr>
</thead>
<tbody>
<tr>
<td>ESET-NOD32</td>
<td>35404</td>
<td>0.400314294657568</td>
<td>0.341918234821158</td>
<td>0.3762688582572053</td>
</tr>
<tr>
<td>AntiVir</td>
<td>28916</td>
<td>0.3857174313986034</td>
<td>0.35360439167922664</td>
<td>0.37249441504356595</td>
</tr>
<tr>
<td>Fortinet</td>
<td>11309</td>
<td>0.3076218211291671</td>
<td>0.45692378455436966</td>
<td>0.3690991001866034</td>
</tr>
<tr>
<td>Ikarus</td>
<td>11605</td>
<td>0.34531932780427554</td>
<td>0.3966825931148037</td>
<td>0.3664689076380225</td>
</tr>
<tr>
<td>VIPRE</td>
<td>34160</td>
<td>0.35927998509018105</td>
<td>0.32120627979087146</td>
<td>0.34360257702575947</td>
</tr>
<tr>
<td>AegisLab</td>
<td>19081</td>
<td>0.21864251575511276</td>
<td>0.5188919016693916</td>
<td>0.3422746158374629</td>
</tr>
<tr>
<td>TrendMicro-HouseCall</td>
<td>13272</td>
<td>0.3429611180911511</td>
<td>0.34099196900334583</td>
<td>0.3421502919961724</td>
</tr>
<tr>
<td>DrWeb</td>
<td>6334</td>
<td>0.3073604111923156</td>
<td>0.3901247352265287</td>
<td>0.341439837358151</td>
</tr>
<tr>
<td>Sophos</td>
<td>7423</td>
<td>0.289326261661417</td>
<td>0.3868687818049352</td>
<td>0.3294908287793363</td>
</tr>
<tr>
<td>AVware</td>
<td>8767</td>
<td>0.237743111177693427</td>
<td>0.43153358820021026</td>
<td>0.3175391903041656</td>
</tr>
<tr>
<td>Comodo</td>
<td>6424</td>
<td>0.2879249629918774</td>
<td>0.3424215248144691</td>
<td>0.31036472374235635</td>
</tr>
<tr>
<td>F-Prot</td>
<td>9266</td>
<td>0.2645551939252682</td>
<td>0.37216099621172916</td>
<td>0.3088634654549874</td>
</tr>
<tr>
<td>McAfee</td>
<td>6361</td>
<td>0.27198170342155936</td>
<td>0.3584585881352673</td>
<td>0.30758983242132143</td>
</tr>
<tr>
<td>McAfee-GW-Edition</td>
<td>5371</td>
<td>0.2609017205111164</td>
<td>0.3433606928224435</td>
<td>0.2948554149922511</td>
</tr>
<tr>
<td>AVG</td>
<td>8876</td>
<td>0.2538577667579801</td>
<td>0.3428874138427105</td>
<td>0.2905170332046338</td>
</tr>
<tr>
<td>NANO-Antivirus</td>
<td>9384</td>
<td>0.2705734506205292</td>
<td>0.29343814688720826</td>
<td>0.27998832555386766</td>
</tr>
<tr>
<td>Baidu-International</td>
<td>5203</td>
<td>0.2571081797779555</td>
<td>0.1967620955397713</td>
<td>0.23225979215046794</td>
</tr>
</tbody>
</table>
Datasets
Goodware and Malware
Because... size matters!
We have a mega database of Android Apps with its market data associated and metadata of the APK itself... and the results in VirusTotal of the apps. It is used in intelligence and researching...

611,323 of our apps were found in VirusTotal, from a total of 742,344 that we got (about September 2014), extrictly from Google Play. Nowadays we have about 3M
04 All together now

What we have done so far. Any malware set may be ok depending on what you need

• We have created a criterion to improve the generation of malware sets based on information retrieved from Google Play.
• It is very customizable.
• This may improve researches about “detecting malware/adware” with machine learning techniques and Android.

For us, goodware dataset is:

• No detections.
• 4.3 stars or even better.
• 70,000 or more downloads.
• More than a month in Google Play.
05
Machine Learning (SVM)
Detecting features and classifying malware
(Research in progress)
05 Target: Realistic and improved classifier


• Machine learning (ML) is a “commodity”. There are so much libraries (SPARK/MLLIB, LIBSVM,...) ant it is easy to analyse data using ML.

• We do not need only goodware/malware sets, but “features” of the apps to build different machine learning classifiers. Which are the bests features?
Previous researches...

- With our huge database (around 3 million apps), we validate previous studies that use the number of permissions (apps) as a feature (goodware/malware).

- Malware does not have more permissions than other apps (in general)...
- Malware is (only) more likely in a range from FOUR to THIRTEEN permissions.

Is the number of permissions a good feature to classify malware?
• Specific combinations of permissions or specific permissions by category (apps) are useful to detect malware?
Features based on permissions...
Let’s try to build a classifier using permissions as the only feature

• Well, the only way to know it is testing... what will happen?
• First: let’s try to propose an app definition only by its permissions.

• It does not seem a very accurate way of defining an app, but this is an experiment to see if it is enough for classifying only by this parameter
• Second: we make a supervised learning process from the apps in both sets (goodware/malware), just taking the permissions into account.
• We have trained a machine learning system with supervised learning implementing Support Vector Machine (SVM) algorithms.

• With our huge dataset (apps) from Google Play and their criteria (previous studies) for choosing a malware set, we were not able to guarantee an acceptable accuracy to distinguish and classify adware/malware and goodware relying only on the permissions.

• Results:

  Sorry... but as far as we know previous researches don’t work with a huge dataset

• What do we do now?
05 **Features based on permissions by category...**

Come on! Let’s try...

- With subsets by category (apps) the classifier better detects goodware/malware than when it uses features based on permissions with general subsets (in this last case we are not able to distinguish between goodware and malware)

<table>
<thead>
<tr>
<th>Category</th>
<th>Accuracy %</th>
<th>Category</th>
<th>Accuracy %</th>
<th>Category</th>
<th>Accuracy %</th>
<th>Category</th>
<th>Accuracy %</th>
</tr>
</thead>
<tbody>
<tr>
<td>Book and reference</td>
<td>77.74</td>
<td>Game arcade</td>
<td>72.57</td>
<td>Lifestyle</td>
<td>74.77</td>
<td>Shopping</td>
<td>86.11</td>
</tr>
<tr>
<td>Business</td>
<td>76.06</td>
<td>Game educational</td>
<td>66.88</td>
<td>Game action</td>
<td>66.66</td>
<td>Social</td>
<td>73.66</td>
</tr>
<tr>
<td>Communicaton</td>
<td>78.33</td>
<td>Game family</td>
<td>71.33</td>
<td>Game sports</td>
<td>68</td>
<td>Sports</td>
<td>77.86</td>
</tr>
<tr>
<td>Education</td>
<td>71.22</td>
<td>Game puzzle</td>
<td>84</td>
<td>Photography</td>
<td>76</td>
<td>Tools</td>
<td>66.33</td>
</tr>
<tr>
<td>Entertainment</td>
<td>66.66</td>
<td>Game Racing</td>
<td>74.92</td>
<td>Personalization</td>
<td>80.26</td>
<td>Productivity</td>
<td>77.40</td>
</tr>
</tbody>
</table>

Without Category (general dataset): Accuracy = 50%

- Although, the results are bad, this study gives us a good idea to move forward: **What will happen if we use more features and specific categories (granularity)?**
What will happen if we use more features and specific categories (granularity)?

Additional specific features could work better in a specific dataset.
What features we use? “Meta-info features”

Let’s try to improve the classifier with some more features.

Do these features are useful to classify goodware and malware?
How to select features?... We will be honest! :D

- Basic statistics and “intuitions”... and, recently, features selection algorithms
Classifying with more features (meta-info)...

Example dataset goodware/malware

- Trend: More features and specific categories (granularity)
- Goodware (True Positive), Malware (True Negative), SVM (Support Vector Machine)
- **10 features:** number permissions, size, certificate, info description, ...
- Results are much better than with features based on permissions only...

- Train: 16,470 (apps goodware) + 8,236 (apps malware)
- Test: 8,236 (apps goodware) + 4,118 (apps malware)
- In total (“global” classification): 37,060 apps = 24,706 goodware + 12,354 malware

**Accuracy = 91.015555% (11,244 apps ok predicted / 12,354 apps to predict)**

- Recall = True positive rate = 90.2507% (measure the proportion of GOODWARE which is correctly identified as such) TP/TP+FN
- True negative rate = 92.9468% (measure the proportion of MALWARE which is correctly identified as such) TN/TN+FP

**Precision = Positive Predictive value = 97% (how good is predicting goodware) TP/TP+FP**

**Negative predictive value = 79.0432% (how good is predicting malware) TN/TN+FP**

- F1 score = f(precision, recall) = 93.5042%

- For us it is a good result as a complementary method to rank apps, especially, when you try detecting malware in a huge dataset, as the real world, and you do not know how to prioritize apps.
05 What will happen if we use more features and specific categories (granularity)?

- For some categories the results are better than the “global” classification (previous). Some examples will be showed later!
- For some categories the results are worse than the “global” classification
  - There aren’t enough samples to train/test... is a problem! How to solve it?
  - A good balance between general & specific dataset it is necessary to get an accuracy classifier.

General Dataset

<table>
<thead>
<tr>
<th>Features</th>
<th>Features based on permissions &amp; general dataset</th>
<th>Classifier</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>GOOD &lt; BETTER</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BAD accuracy!!!</td>
</tr>
<tr>
<td>Meta-info features &amp; general dataset</td>
<td>Features based on permissions &amp; Specific dataset</td>
<td>Features</td>
</tr>
</tbody>
</table>

Specific dataset (Category)

The same features for goodware/malware dataset
How to improve the classifier...

• We are searching for new features...

• We will be honest, the classifier is good but... I would like to improve the results analysing code... sometimes is the only way to detect malicious code with high accuracy.

• But... analysing code is a slow process!!!

• Are there any solutions to analyse code quickly without reverse engineering?

• Could it be possible to add this kind of solutions as new features to our classifier?

• Why not detecting patterns analysing .DEX files and Android bytecodes (opcodes) directly?

Crazy idea?
06

Machine Learning and Android bytecodes

Analyzing code without "reverse engineering or code execution"

(Research in progress)
Analyzing DEX (Dalvik-bytecode)...

For the same datasets (goodware/malware) used “before”...

Example: NOP 0x00h, ADD 0x90h,...
Analyzing DEX (Dalvik-bytecode)...
For the same datasets (goodware/malware) used “before”...

• Why these features? Intuitions & “some” previous researches...

  Using opcode-sequences to detect malicious Android applications

  IEEE ICC 2014

• We collect new features per each app:
  • N-gram opcodes (is a contiguous sequence of n opcodes in each .dex file)
  • Number of Opcodes (number of instructions)
  • Size of code section
  • % per each type of Opcode
  • % per each instruction format (30 formats)
    • https://source.android.com/devices/tech/dalvik/dalvik-bytecode.html
  • % per groups of instructions (13 formats)
    • Conditional, transfer, flow, arithmetic, moves, literals...
  • Entropy and Entropy per “blocks of N instructions”

• Does Malware have less code? Is it obfuscated? Opcodes pattern helps to distinguish malware? Developer’s programming style?
Classifying using Opcodes...

For the same datasets (goodware/malware) used “before”...

- These new features do not classify malware better than our previously used meta-info features (permissions, size, description, certificate...) but, both combined per category they do!!!

- **Category classification (metainfo features + opcodes features):** We improve the previous results with opcodes patterns (we have again problems with some categories if there are not “enough” samples)

![Diagram showing classification process with datasets, features, and classifiers]

**GOOD……… < BETTER < MUCH BETTER**
**Category classification...**

Metainfo features + Features based on opcodes...

- We are getting a very good results per category...
- Example: BOOK_AND_REFERENCE category
  (Remember: with features based on permissions only → Accuracy=77.74% (for this category)

<table>
<thead>
<tr>
<th>“Global Classification Features based on META-INFO”</th>
<th>Features based on META-INFO BOOK_AND_REFERENCE</th>
<th>META-INFO + OPCODES PATTERN Categ: BOOK_AND_REFERENCE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy = 91.015555%</td>
<td>Accuracy = 96.26 %</td>
<td>Accuracy = 97.06 %</td>
</tr>
<tr>
<td>Recall = 90.2507% (true positive rate)</td>
<td>Recall = 96.0937% (tpr) [tnr = 96.6386%]</td>
<td>Recall = 96.8627% (tpr) [tnr= 97.5%]</td>
</tr>
<tr>
<td>[true negative rate = 92.9468%]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Precision = 97% (positive predictive value)</td>
<td>Precision = 98.4% (ppv) (npv = 92%)</td>
<td>Precision = 98.8% (ppv) (npv=93.6%)</td>
</tr>
<tr>
<td>[negative predictive value = 79.0432%]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>F1 score = 93.50%</td>
<td>F1 score = 97.2321%</td>
<td>F1 score = 97.8217%</td>
</tr>
</tbody>
</table>

- We proved, with different categories/datasets, that is possible to improve the classification results with opcodes... It is difficult to improve more by category :D
07
Conclusions
Conclusions

• Analyzing mobile malware is expensive. With 3k-5k new apps uploaded daily, it may be a good idea to prioritize efforts filtering by suspicious apps first before deep analysis.

• Classifying apps without recurring to traditional ways, is usually based on characteristics of the app and Machine Learning classification. But this strongly depends on:
  - The malware/goodware set you choose, because:
    - You may have a “not big enough” set.
    - Not big enough set of characteristics so the apps in the set are properly “profiled”.
    - Not being very good at determining which is malware or not (VT dependency).
  - And what “features” you chose to define an app...
  - And what you consider as an “app” (take advantage of its circumstances)

• We discover a lot of features (meta-info and opcodes features) useful to classify adware/malware and goodware. We have a good detector, not relying on code, that offers accurate and high performance results.
Opcodes in Google Play
Tracing malicious Applications

Speakers

Alfonso Muñoz, PhD (@mindcrypt)
Senior Cybersecurity Researcher – alfonso.munoz@11paths.com

Sergio de los Santos (@ssantosv)
Head of Lab in ElevenPaths • ssantos@11paths