CLASSIFYING PUAs IN THE MOBILE ENVIRONMENT

Vanja Svajcer & Sean McDonald
Sophos

Email {vanja.svajcer, sean.mcdonald}@sophos.com

ABSTRACT

The issue of PUAs (Potentially Unwanted Applications) in the world of desktop sample processing is well understood. Typically, we classify as 'potentially unwanted' executables which are borderline malicious but which may in some cases provide certain benefits to the end-user. These applications are assigned to one of the predefined PUA categories, giving the user the option to manually authorize their usage.

Has the world of PUAs changed with the advent of mobile apps? As the revenue model for application developers changes, should the security industry apply different criteria when considering potentially unwanted mobile applications?

In mid 2013, there are over 700,000 apps on *Google Play* and over 800,000 apps on *iTunes*, with numerous alternative application markets serving their share of *Android* apps. The major source of income for most of the app developers is advertising revenue realized by integrating one or more of the advertising frameworks.

The difference between malware, PUAs and legitimate apps for mobile platforms is often less clear than it is within the desktop world. We have seen several cases where not even security vendors agree on how to classify apps containing multiple advertising frameworks such as Plankton or Newyearl-B. This causes both application developers and the developers of individual advertising frameworks some confusion as to which features are acceptable.

This paper introduces a structured PUA taxonomy for mobile apps which can be applied both by security vendors and by mobile app developers. Wherever possible, we use categories closely related to desktop PUAs and introduce new ones that are particularly relevant to mobile environments. We apply the categorization to an existing corpus of mobile PUA samples, legitimate apps and individual advertising frameworks.

INTRODUCTION

Potentially Unwanted Programs/Applications (PUPs/PUAs) are not a recent phenomenon. This general class of application, distinct in name and meaning from both malware and clean and trusted applications, evolved because:

- Digital advertising grew to become a principal source of revenue.
- In trying to maximize revenue, some authors of applications displaying digital advertisements pushed the boundaries of what was acceptable to the end-user.
- End-users required protection against such applications because, whilst not strictly malicious, some were installed without consent and/or were very difficult to remove.
- Security software vendors and/or start-ups in the field of application reputation filtering had to tread carefully through a litigation minefield where the legal fraternity were still catching up on what risk such applications actually posed to end-users and productivity at large.

What emerged was a general acceptance that certain applications are potentially unwanted. Whilst the major category came to be called 'adware' (although that term is also used in other contexts), other categories of PUAs/PUPs also emerged because the type of application could aid malware in the infection process, cost users money, or simply could collect too much information and affect users' privacy.

At one point, another term commonly used to refer to malicious programs intending to monitor a victim's machine, 'spyware', became synonymous with PUAs, even if a set of all PUAs encompassed many other subcategories.

These other categories include, but are not limited to:

- Remote administration tools
- Remote monitoring tools
- Hacking tools
- Premium rate dialler tools.

As the market matured, and users' tolerance for what was even potentially unwanted decreased, the number of PUAs/PUPs declined as the developers diversified to clearly legitimate and clearly malicious.

Certain categories became accepted as simply unwanted (malicious), whilst others were replaced with a new breed of application that started offering similar services as in the past but did so in an open and unobtrusive fashion (legitimate).

However, users still desired a means to control such applications – not because they were malicious, but for performance/productivity/general security reasons. This is how the concept of controlling installed and executed applications emerged (application control). One might ask why this history lesson is relevant. We often hear that history tends to repeat itself and we shall see it in the case of mobile PUAs.

MOBILE PUA HISTORY

Whilst much of the story can also be applied to other mobile platforms, we have used *Android* as the basis of our work for this paper.

The first app that should have been classified as a PUA from the outset was Plankton. The applications that included the Plankton (Apperhand, Startapp) framework were published on the *Google Play* market for more than two months before anybody noticed anything unusual about them. Some of the applications became very popular and were downloaded over 100,000 times.

We investigated the functionality of Plankton's available samples at the time and concluded that whilst the majority of existing *Android* malware samples were quite obvious in their intention to provide some kind of benefit, usually financial, to the attacker, we could not say outright that there was a malicious intention with Plankton.

Plankton's code suggests that it is an advertising platform, but it does not disclose its purpose at installation. After security vendors decided that Plankton should be classified as malicious, the *Google Play* security team removed the apps from the *Google Play* market.

With the appearance of advertising frameworks competitive to *Google*'s Admob, developers of the frameworks, as well as application developers, have been looking for more successful methods to increase the revenue through new types of ad placement.

Some of the *Android* features lent themselves to a set of increasingly intrusive functionality, such as push adverts, displaying messages outside of the application context, placing shortcuts to advertised services on the home screen, changing mobile browser bookmarks, and more serious issues involving data being sent to a centralized location which was later presumably used for improving the targeting of adverts.

There came a point at which some of the new advertising functionality became too intrusive. It started to seriously affect user experience and prompted users to look for help with app classification, which was a part of the standard scope of security software.

ADVERTISING FRAMEWORKS FOR ANDROID

As part of the preparation for writing this paper, the authors have investigated and worked on the implementation of several popular advertising frameworks into a test app which has never been published to an application market. The test application would be classified as a PUA under several criteria described in the proposed PUA taxonomy.

Motivation behind the implementation of advertising frameworks

Advertising frameworks have played a significant role in the development of apps since the inception of the concept of so-called 'free' apps. The concept of free apps is enticing to end-users as it often allows them to enjoy fully functional apps without having to pay any money up front. However, the payment is often just postponed and it happens through other means, either through in-app purchasing or through displayed adverts, data leakage and loss of privacy.

If developers want their apps to be successful – which means having hundreds of thousands of installs – they have no choice but to offer their app first as 'free'. Only later may they be able to capitalize on their install base through the income made by selling the app directly and removing the otherwise necessary advertising burden.

When considering the advertising ecosystem, we should be aware of several actors: advertisers, mediators, developers and users. The security industry is usually concerned only with the last category, but we should be very careful not to infringe on other actors' rights to reach their goals. For advertisers, the goals are lead generation and product promotion; for mediators, income from adverts placed by advertisers; for developers, income made from integrating advertising SDKs in their apps.

Performance indicators

Depending on the actor, several advertising performance indicators are used to allow the actor to track the success of their advertising or monetizing strategy. Although different advertising frameworks each use their own key performance indicators (KPIs), it seems that there is a general consensus that the common KPI for all frameworks is eCPM (see Table 1).

eCPM stands for Estimated Cost per Mille (thousand). What is cost to advertisers is revenue for developers and is measured per thousand impressions of an advert type in apps integrating a particular advertising framework. It is interesting to look at the range of eCPM values for different advertising frameworks and different types of ads (see Table 2).

AD TYPES AND ASSOCIATED eCPM RATES

Banner ads, displayed inline with other elements of the application activity, usually yield the lowest eCPM rates – typically in the range of $0.02 to $0.10, which means that a popular app, with a million daily active users, may make $20 to $100 a day by displaying them.

Interstitial ads are often displayed between different game levels and take up most of the available screen area. They have much higher eCPM rates, reportedly ranging from a relatively low $0.5 to a very high $10. Despite their higher success rates, interstitial ads degrade user experience and are not good for the reputation of an app.

Push or notification ads, which display adverts in the OS notification area, infringe on *Google*'s Developer Content Policy if they do not specify which app is responsible for placing them there. Under our current criteria, ad frameworks using push notification are classified as PUAs. The most popular push frameworks are Airpush and Leadbolt. Reported eCPM rates for push apps are very similar to interstitial ad eCPM rates.

Icon ads are advertising units that place one or more shortcuts on the device home screen. Shortcuts usually link to web pages that advertise other apps or link directly to *Google Play* URLs that suggest the user installs the advertised app. Icon ads reportedly have high eCPM rates, similar to interstitial and notification ads, but are also very confusing to the end-user and significantly downgrade both the user experience and the reputation of the apps that use them.

	Impressions	Clicks	eCPM	RPM	RPMD	Pushes	EPC	CTR	Fill rate
Admob			x	x					x
Leadbolt	x	x	x				x	x	
Airpush	x		x			x			
Applovin	x	x	x						
Mobilecore			x						
Startapp			x		x				

Table 1: Popular advertising frameworks and their KPIs.

	Banner	Interstitial	Push	Icon	Other
Admob	x	x			
Leadbolt	x	x	x	x	Multimedia
Airpush		x	x	x	Multimedia
Applovin	x	x			
Mobilecore		x		x	Slider
Startapp		x (on exit)		x	Browser search, bookmarks

Table 2: Advertising framework and ad units.

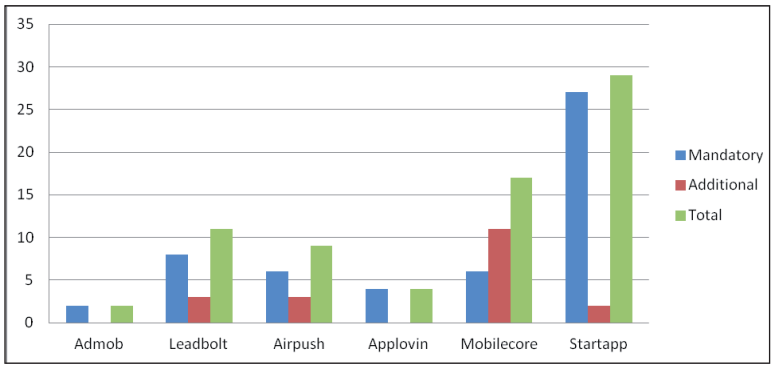

Figure 1: Number of permissions for analysed advertising frameworks.

Permissions

In addition to the degradation of user experience caused by the placement of adverts, advertising frameworks typically lower the security posture of an app by including additional *Android* permissions not required for the original app functionality. Additional permissions may also lower an app's reputation since many users will question the reasons why certain permissions have been requested.

When looking at the number of permissions required by different advertising frameworks (Figure 1), we could state that the number of required permissions is inversely proportional to a framework's reputation.

Permissions often used by advertising frameworks and affecting user experience the most are:

- Location based
  - android.permission.ACCESS_COARSE_LOCATION
  - android.permission.ACCESS_FINE_LOCATION
  - android.permission.ACCESS_LOCATION_EXTRA_COMMANDS
- Degrading user experience
  - android.permission.WAKE_LOCK
  - android.permission.RECEIVE_BOOT_COMPLETED
  - com.android.launcher.permission.INSTALL_SHORTCUT
  - com.android.launcher.permission.UNINSTALL_SHORTCUT
- Affecting privacy
  - android.permission.GET_ACCOUNTS
  - android.permission.READ_PHONE_STATE
  - com.android.browser.permission.WRITE_HISTORY_BOOKMARKS
  - com.android.browser.permission.READ_HISTORY_BOOKMARKS

Framework reputation

When considering individual advertising frameworks, one should also consider their reputation with end-users and developers.

A good indicator of the reputation of individual frameworks with end-users is a relative ratio between the share of apps integrating an advertising framework as a proportion of all apps published on the *Google Play* market and the share of actively installed apps using the framework as a proportion of all installed apps. For this indicator, authors have used data maintained by an alternative *Android* market site, Appbrain.com.

We can define the reputation indicator as:

$$Rep = \frac{Apps_{installed}\,\%}{Apps_{implemented}\%}$$

The higher the reputation indicator, the stronger the user reputation of the framework. It is fair to say that any advertising framework should aim to have a user reputation indicator greater than 1 (see Figure 2).

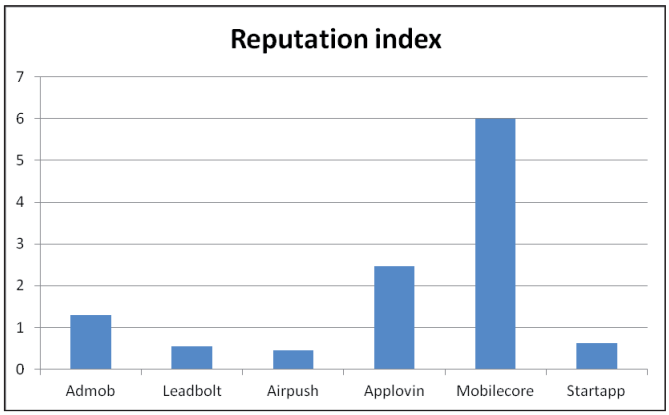

Figure 2: Application reputation index.

EXTERNAL FACTORS AS MOTIVATORS FOR MOBILE PUA CLASSIFICATION

International legislation as a source of classification criteria

Understanding the privacy laws of any single country/jurisdiction requires the careful interpretation of laws by a seasoned professional and within the wider system of law applicable. Things get even more complicated because of the cross-border nature not only of data collection but of subsequent transfer of harvested data across the world. As such, a couple of researchers cannot possibly ascertain the current legal position of international laws detailing data privacy. However, that does not mean we cannot glean some big-picture themes that may materially impact how apps could be classified, or should be classified at some future point (Figure 3).

Figure 3: Tag cloud showing importance of concepts in legislation by word frequency.

There has been a growth in the enactment of privacy laws around the world (see Figure 4). Certainly, such laws cover more than just the privacy concerns of digital/online data – some cover only the practices of publicly run entities and some only the practices of privately run companies. Nevertheless, there is a trend towards greater regulation of the collection and usage of data that is considered personal and private, which provides arguments that privacy laws, and adherence to them, can form the basis for classifying apps as PUAs.

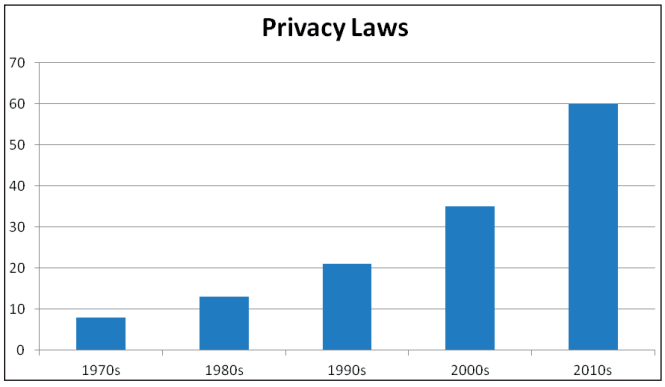

Figure 4: International growth of legislation regulating privacy.

Smartphones and tablet devices raise new privacy concerns that are not even present in the online space: such devices regularly contain a user's name, phone number, personal contacts, SMS message history, service carrier information, photographs and other multimedia, location information, installed apps and browser settings, and history. Certainly, the misuse of specific data such as a person's name or SMS message history could pose a privacy issue on its own. However, the accessibility of other, less specific data when used in combination could create a fairly detailed profile of a user. It could be argued that an app that harvests too much personal data on its own, or in combination with an advertising package exhibiting data-harvesting functionality, should be declared at least a PUA (if not malware) under the category of 'Privacy Concern'. The basis for such a classification would be that the collection of personal data contravenes a domestic or internationally recognized personal privacy law or initiative.

There are many domestic laws implemented around the world, and issues such as extraterritoriality and transaction locality within the connected world are out of the scope of this paper (and the researchers writing it!). However, there are two models that are proving influential and their basic principles and core differences should be discussed in more detail. The Working Party set up under Article 29 of the EU Data Protection Directive (95/46/EC) recently reviewed the legal framework that applies to apps. The big themes from this review are:

- That app developers are probably unaware of data protection requirements and the risks that lack of transparency and meaningful consent to data collection and usage pose to a user.
- Poor security measures are being used to collect, store and transmit personal data,which is a privacy concern in its own right.
- There is an apparent trend towards data maximization, with some apps and ad libraries collecting more data than is strictly necessary to offer the proposed service.
- The huge issue of consent: what constitutes consent? When should consent be requested? User acceptance to install an app does not necessarily imply consent to the collection of personal data and requirements of consent can change depending on age (for example, children may require a parent's consent).
- Low-entropy device identifiers should be used as opposed to identifiers such as IMEI which may provide the potential to track users across apps and stitch together profiles in ways that are unforeseen to the user.

These big themes seem very much in favour of protection of the user. The US Consumer Privacy Bill of Rights also recognizes that privacy is not an outmoded value. However, it seems to favour a multi-stakeholder approach to creating codes of conduct. The US model, with private sector participation being voluntary, seems to lean much more towards protecting (American) industry and innovation, and allowing market forces to determine guidelines. There is certainly merit in the argument that market forces, including consumer demands and choice to only use services certified to meet a certain set of requirements, can more quickly result in established guidelines. However, a counter-argument could be that laws make unwanted behaviour a crime and play an important role in deterring such unwanted behaviours. The big themes from the US model are revolving around consumers' rights to:

- Individual control – consumers should be offered simple choices, and at a meaningful time, on how their data will be collected, used and disclosed to others. This includes withdrawal of consent. This model also puts the user in the middle of control by advocating that the user has the responsibility to exercise choice in available privacy settings to begin with.

- Transparency – consumers should be informed of what personal data is being collected, why it is needed, how it will be used and subsequently deleted, and whether the data is shared with third parties. Notifications should take account of smaller screens on mobile devices and present the most relevant information in a way that is meaningful on such devices.
- Respect for context – personal data should only be used and disclosed for purposes consistent with the original user consent. Age and familiarity with technology are also important elements of context.
- Security – reasonable safeguards should be enacted to control risks such as data loss, unauthorized access and improper disclosure.
- Access and accuracy – personal data collected should be accurate and there should be a reasonable means for people to access the data collected on them.
- Focused collection – only as much personal data as needed to implement the service should be collected (more data should not be collected opportunistically).
- Accountability – there should be means to ensure that a company is adhering to the measures in the Consumer Privacy Bill of Rights.

There are certainly some common themes among the two models: recognition that unique device identifiers are personal data, that poor security practices in handling personal data is a privacy risk, that consent requires users to be given meaningful information at appropriate times so they can make informed judgments, that age and familiarity with technology are important factors in how consent is requested, and that only enough data as needed should be collected.

These two models created by two economic powerhouses are likely to influence other governments and jurisdictions in how they tackle personal-data privacy concerns. Putting aside the different philosophies on achieving compliance, there is a substantial overlap in the big themes. It could be argued that an app or ad package/library that collects personal information in a manner that is inconsistent with these ideals is at least a PUA. We also believe it could be argued that a new PUA category simply named 'Privacy Concern' to cover all such apps would allow the security industry to play its role as a multi-stakeholder in bringing privacy concerns to the forefront of the market, which might drive user demand and encourage app developers to incorporate these enhanced privacy standards.

Google developers application content policy

While investigating reference material for our paper, we have considered several other papers and documents and decided that *Google*'s Developer Content Policy for *Google Play* already lists several criteria which can be used when considering a PUA taxonomy not just for *Android* but for all mobile platforms.

Testing organizations as a driver behind PUA classification

It is not only the providers of security solutions that are having a difficult time classifying apps. Testing organizations are in a unique position because they are certifying the accuracy of a particular solution's detection rate. To be meaningful, a testing organization must first categorize apps to be tested in an environment where there are yet no agreed definitions encompassing the range of app behaviours seen today.

Testing organizations play a meaningful role in validating and certifying the abilities of different solutions. However, in regards to apps, especially given the steep growth of not only PUAs but of malware in this space, there is the question of how testing organizations classify apps to be tested and whether that classification goes beyond simply malware-versus-PUA. For example, if the tester's own analysis teams are drawing the line between which samples are malware versus PUAs, then testing companies really are playing a major role in defining what PUAs are, and as such need to publish reasons for their decisions for wider scrutiny and influence. If, however, testers are relying on a single/few trusted sources, then the obvious question is what makes those sources trusted so that they can play a wider role in defining app categories/classifications? If testers are using a fairly wide and representative set of detection results to collect samples for testing, then testing organizations can play a significant role in supporting a market-based solution to the classification problem.

A philosophy of the US Consumer Privacy Bill of Rights is that a multi-stakeholder process can produce solutions quicker than legal regulation or international treaties. Assuming there is at least some merit to this suggestion, it could be argued that testing organizations are in a good position to coordinate a definition/classification agreement among the many solution providers. It has been our experience that testing organizations have been very receptive to challenges on the inclusion of samples in a test, so there is already likely to be an element of this happening on a per test basis. What is missing is the publication of reasons for the inclusion or subsequent exclusion of samples over time that could result in a widely accepted set of definitions/classifications based on the industry feedback and cooperation.

Testers also have a role to play in the validation of classifying an app as a PUA. It has been stated that app authors may be unaware of app development best practices and so collect more information or share it in ways unknown to the end-user. Testing organizations validating a vendor's classification of an app as a PUA could assist in urging the app authors to adopt better development practices and/or choose more end-user friendly, and widely recognized as legitimate, ad packages/libraries.

CLASSIFICATION GUIDELINES AND PUA TAXONOMY

This section is split into two parts. The first part introduces current mobile PUA categories as used today by *Sophos* researchers, while the second part proposes additional categories which should be considered when putting together an all-encompassing mobile PUA taxonomy.

When we looked at the functionality and features of mobile PUAs, we concluded that there are two major groups of apps to classify.

The first group contains apps whose principal function warrants classification into one of the associated mobile PUA categories. The second group contains all types of apps, regardless of their intended functionality, which contain either

one of the aggressive advertising frameworks or several advertising frameworks whose overall code size outweighs the benefits of the app's principal functionality. In business terms, these apps have a high cost-benefit ratio and therefore users should be alerted to their presence.

Current classification and mobile PUA categories

Classification based on application functionality

An application irrespective of any explicit End-User Licence Agreeement (EULA) or other offer and acceptance of terms will be classified as a PUA if it:

- Contains functionality to root or jailbreak a device.
- Remotely monitors a user or device – apps to monitor a wife/husband/partner or general device activity fit within this category. Applications that provide genuine device administration and are advertised as such do not.
- Has been cracked or repackaged to avoid purchase – apps that do not exhibit malicious functionality and have been repackaged, for example to avoid payment, fall into this category. Applications that have been repackaged to include extra functionality may also belong to this category. If the app contains malicious functionality, malware detection is advised.
- Abuses a third-party service against terms and conditions – an application that uses/consumes data against/in violation of the terms of the producer of the data belongs to this category. For example, an application that uses *VirusTotal* data as a source of classification data without agreeing terms.
- Is compiled from open source code yet modified to include advertisements.
- Contains more ad packages than useful functionality – apps that contain three or more ad packages fall within this category. Applications containing ad packages that have clearly been reskinned numerous times also fall within this category.

Classification based on ad package/framework

Ideally, apps should not be classified as PUAs simply because they contain an individual ad package: many apps' revenue is built on advertisements, and this is a legitimate method of generating income for app creators.

However, some ad packages can be considered 'aggressive' and any application that contains them may be classified as a PUA. The general line is whether the ad package displays ads only within the application context and not more, or whether the ad package displays ads outside the app. In addition to that, advertising frameworks that change device settings such as browser home or search pages to support ads or funnelling traffic should also be classified as PUAs.

What constitutes an aggressive advertising framework?

An ad package, irrespective of any explicit EULA or other offer and acceptance of terms, may be generically classified as a PUA where the package:

- has the potential, and is often used, to set a new search homepage
- has the potential, and is often used, to add a new bookmark
- has the potential, and is often used, to create a link/icon on the home screen, or
- has the potential, and is often used, to display ads outside the application, or
- has the potential to, and often does, display ads of adult content without regard to audience (for example, a package dedicated to serving porn advertisements would fall within this rule)

or

- has the potential to, and often does, collect and submit potentially identifiable information connected with the user or the device, such as location, IMEI, IMSI, contacts or SMS messages.

What constitutes an acceptable advertising framework?

An ad package from a well-known vendor (that has a natural obligation to act responsibly), or a lesser-known ad package using common sense, which plays 'fair' and which:

- displays a EULA which clearly states what types of adverts will be used and which data may be shared with the developers of the advertising framework and third parties
- only displays ads within the application
- only displays ads containing content suitable and directed to a target audience

may be considered as acceptable and not classified as a PUA.

Initial mobile PUA categories

Remote administration tools (Spyware)

A myriad of apps are produced and designed to allow monitoring of activities of the target cell phone device from a remote location. Those monitoring activities may include, but are not limited to: reading SMS messages, reading email, tracking GPS location, monitoring Internet use, accessing the user's calendar and address book, and using the phone and camera to record audio and video for the purpose of user surveillance.

Although developers often state that their intended purpose is legitimate, spyware programs represent a big concern for many users, especially those with a lower level of computing awareness.

Users often assume that there is a piece of spyware installed on their devices without their knowledge, even if they do not have a valid argument as to why that may be the case. These users are often looking for reassurance from their chosen security app.

Good examples of remote administration tools are *Droid Commander* and *Mobi Spy*. When detecting remote administration tools and any apps which are classified on the basis of their functionality, it is recommended that real app names are used in detection reports.

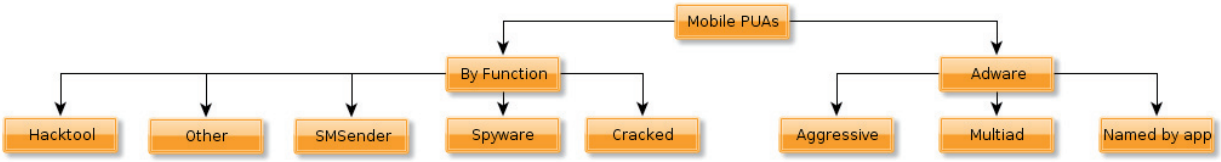

Figure 5: Current mobile PUA taxonomy.

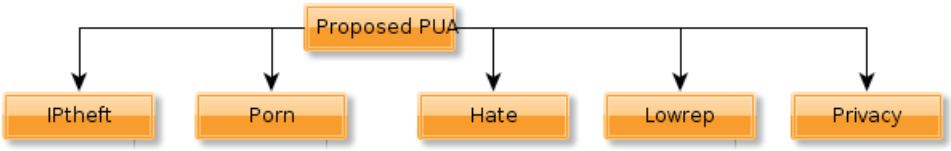

Figure 6: Proposed additional mobile PUA categories.

Cloned and modified apps (Cracked)

The category of cracked apps is already well known in the desktop world but takes on a whole new dimension in the *Android* environment. Due to the nature of the development tool chain translating Java class file to the *Android* executable bytecode format (Dex), it is trivial to reverse engineer and modify unobfuscated executables to remove licence checks or include additional functionality. Many crackers exploit this fact to remove often the most rudimentary software protection then post the cracked apps on app-sharing sites.

Due to the standard practice of using self-signed certificates, it is very easy to clone an app and include additional functionality such as SMS sending code or advertising frameworks designed to drive the revenue to the non-legitimate developer of the app.

Ad-supported apps (Adware)

Ad-supported apps are all apps classified based on the guidelines outlined in the section 'Classification based on ad package/framework'. When detecting ad-supported apps based on aggressive advertising frameworks, it is recommended that the name of the detected advertising framework is used in the detection report, for example Airpush or Leadbolt.

When detecting other categories of ad-supported apps based on the presence of multiple advertising frameworks affecting user privacy or degrading user experience, it is recommended that a generic detection name is used, which lets the user know that the app is detected based on the presence of multiple advertising frameworks, for example 'Android MultiAd'.

In some cases, like Newyearl-B, the principal motivation of the app's developers is both to misrepresent its functionality and to include multiple advertising frameworks. In similar cases, it is recommended that a detection name is used which uniquely identifies the app, rather than the implemented advertising frameworks.

An additional discussion of the ad-supported apps and advertising frameworks is given in the section 'Advertising framework for *Android*', as understanding of the mobile app ecosystem is crucial when making a decision about which mobile advertising frameworks should be classified as PUAs.

SMS-sending applications (SMSender)

SMS-sending applications are not outright malicious since their principal functionality may in some cases be useful to the end-user. For example, SMSenders may send SMS messages in the background, potentially as a part of a licensing scheme.

In some countries, especially in Russia, several alternative *Android* markets practise adding some kind of 'licensing code' that sends SMS messages to numbers that drive the revenue to site owners.

Hacking tools (Hacktools)

The hacking tools category encompasses programs that can be used to assist in gaining entry to a network, mobile device or software program. These are sometimes used by attackers but can also be used legitimately for assessing network security. The presence of hacking tools on an enterprise network without the knowledge of the administrator may indicate that an attack is in progress, and this is the main reason why they should be detected.

Apps that do not belong into any of the above categories (Other)

This is the last category currently used by *Sophos* and it encompasses all apps that could not be classified into one of the previously described categories.

Proposal for additional mobile PUA categories

Sexually explicit apps (Porn)

Apps that contain nudity, graphic sex acts or sexually explicit material should be categorized as a PUA category.

Violence promoting apps (Hate)

Apps that contain materials which threaten, harass or bully other users as well as apps that promote hatred toward groups of people based on their race or ethnic origin, religion, disability, gender, age or sexual orientation/gender identity should be classified in this subcategory.

Apps that infringe on intellectual property (IPTheft)

Apps that infringe on the intellectual property rights of others

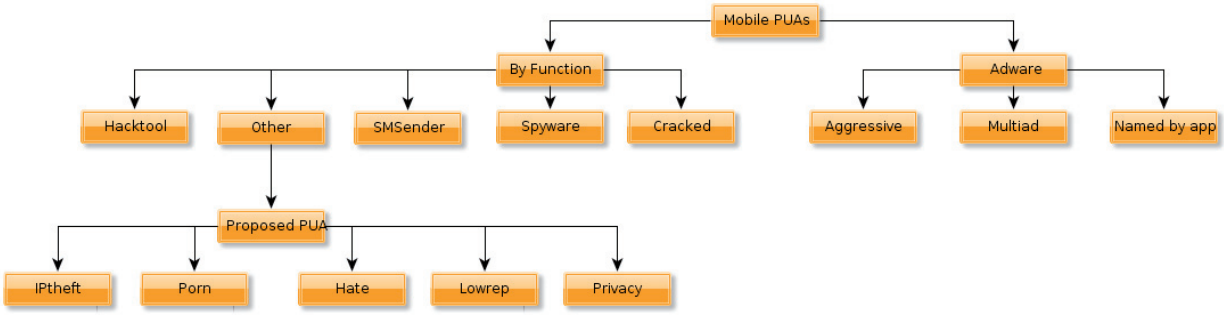

Figure 7: Full mobile PUA taxonomy.

such as patent, trade secret, copyright or encourage or induce infringement of intellectual property rights should be classified in this subcategory.

Apps that affect users' privacy (Privacy)

Apps that leak potentially identifiable user data either with or without the knowledge of the app developer may be classified in the privacy loss category (Privacy). For example, developers may use third-party libraries which may send potentially identifiable user or system information to a location owned by a third party. We have already mentioned in detail that protecting privacy is one of the main areas of concern for users and the area where new legislation is constantly being developed.

Apps that artificially improve their reputation (Lowrep)

Apps with misleading product descriptions in store or apps using keywords that are irrelevant to their actual content in an attempt to manipulate ranking in the search results should be classified as Lowrep.

Applications where developers offer incentives to users to rate an application with higher ratings, as well as apps whose primary functionality is to drive affiliate traffic to a website may also be categorized as Lowrep.

CONCLUSION

Potentially Unwanted Programs/Applications are not a recent phenomenon. However, mobile environments, where developer revenues are largely based on successful implementation of various advertising schemes, represent a significant problem for security companies when considering classification of applications as PUAs.

This issue is also recognized outside of the security industry, with many countries working on guidelines and legislative documents concerning mobile app developers and practices acceptable for end-users.

The authors of the paper have considered several external and internal motivators for defining PUA classification criteria and have described a PUA taxonomy which may be used as a guideline by security vendors and testing organizations while analysing and classifying apps (see Figure 7).

It is unlikely that we will ever reach a consensus on what constitutes mobile PUAs and which categories should be used for classification. Nevertheless, we hope that this paper will bring us closer to a better common understanding and perhaps even a common solution to the problem.

BIBLIOGRAPHY

[1] Google Play Developer Distribution Agreement. https://play.google.com/about/developer-distribution-agreement.html.

[2] Google Play Developer Program Policies. https://play.google.com/about/developer-content-policy.html.

[3] Mobile App Advertising Guidelines. https://www.lookout.com/resources/reports/mobile-ad-guidelines.

[4] Stevens, R.; Gibler, C.; Crussel, J.; Erickson, J.; Chen, H. Investigating User Privacy in Android Ad Libraries. MOSTConf Mobile Security Technologies (MOST), 2012.

[5] Grace, M.; Zhou, W.; Jiang, X. Unsafe Exposure Analysis of Mobile In-App Advertisements. Proceedings of the 5th ACM Conference on Security and Privacy in Wireless and Mobile Network, Tucson, Arizona, April 2012.

[6] Hornyack, P.; Han, S.; Jung, J.; Schechter, S.; Wetherall, D. These Aren't the Droids You're Looking For – Retrofitting Android to Protect Data from Imperious Applications. In Proceedings of the 18th ACM CCS, 2011.

[7] Leontiadis, I.; Efstratiou, C.; Picone, M.; Mascolo, C. Don't kill my ads! Balancing Privacy in an Ad-Supported Mobile Application Market. HotMobile 2012, 13th International Workshop on Mobile Computing Systems and Applications.

[8] Kelley, I.; Benisch, M.; Cranor, F.; Sadeh, N. When Are Users Comfortable Sharing Locations with Advertisers? Technical report, CyLab – Carnegie Mellon University.

[9] Gibler, C.; Crussel, J.; Erickson, J.; Chen, H. AndroidLeaks: Automatically Detecting Potential Privacy Leaks In Android Applications on a Large Scale. TRUST 2012, 5th International Conference on Trust and Trustworthy computing.

[10] Shekhar, S.; Dietz, M.; Wallach, D. AdSplit: Separating smartphone advertising from applications. 21st USENIX Security Symposium, 2012 Bellevue.

[11] Google tells ad-blocking utilities on Android: "You're fired!". http://nakedsecurity.sophos.com/2013/03/14/google-tells-ad-blocking-utilities-on-android-youre-fired/.

[12] Plankton malware drifts into Android Market. http://nakedsecurity.sophos.com/2011/06/14/plankton-malware-drifts-into-android-market/.

[13] Fake Plants vs Zombies and other Android games infiltrate Google Play store, make money for fraudsters. http://nakedsecurity.sophos.com/2013/01/21/fake-plants-vs-zombies-android-game/.

[14] What is worse on Android? Malware or PUAs? http://nakedsecurity.sophos.com/2012/09/13/potentially-unwanted-apps-on-android-more-prevalent-than-malware/.

[15] Leadbolt. http://www.leadbolt.com/.

[16] Airpush. http://www.airpush.com/.

[17] Startapp. http://www.startapp.com/.

[18] Admob. http://www.google.com/ads/admob/.

[19] Waps. http://www.waps.cn/.

[20] Joint CDT, FPF Statement on the Development of App Privacy Guidelines. https://www.cdt.org/pr_statement/joint-cdt-fpf-statement-development-app-privacy-guidelines.

[21] EFF Mobile User Privacy Bill of Rights. https://www.eff.org/deeplinks/2012/03/best-practices-respect-mobile-user-bill-rights.

[22] Seizing Opportunity: Good Privacy Practices for Developing Mobile App. Office of the Privacy Commissioner of Canada. http://www.priv.gc.ca/information/pub/gd_app_201210_e.pdf.

[23] Opinion 02/2013 on apps on smart devices. European Union, European Union. http://ec.europa.eu/justice/data-protection/article-29/documentation/opinion-recommendation/files/2013/wp202_en.pdf.

[24] Consumer Data Privacy in a Networked World. The White House. http://www.whitehouse.gov/sites/default/files/privacy-final.pdf.

[25] Making money with Android Forum. Various posts by developers related to implementation of advertising frameworks and their behaviour and performance. http://forums.makingmoneywithandroid.com/advertising-networks/.

[26] Apple App Store review guidelines. https://developer.apple.com/appstore/resources/approval/guidelines.html.

[27] Counterclank is (not) malware. http://nakedsecurity.sophos.com/2012/02/02/android-counterclank-is-not-malware/.

[28] Mobispy. http://www.mobispyapp.com/.

[29] Droid Commander. https://play.google.com/store/apps/details?id=com.netclearance.dclite.

[30] Androguard. https://code.google.com/p/androguard/.

[31] Way of the Android cracker. http://androidcracking.blogspot.com/2012/01/way-of-android-cracker-0-rewrite.html.

[32] Signing your applications. http://developer.android.com/tools/publishing/app-signing.html.

[33] App Annie Index Q1 2013. http://blog.appannie.com/app-annie-index-market-q1-2013/.

[34] Online Advertising Concepts 101. http://enlogica.com/strategy/online-advertising-definition.

[35] Android ad network stats. http://www.appbrain.com/stats/libraries/ad.

[36] Greenleaf, G. Global data privacy laws: 89 countries, and accelerating. 2012. Queen Mary University of London, School of Law Legal Studies Research Paper No. 98/2012.

[37] Article 29 Working Party. http://ec.europa.eu/justice/data-protection/article-29/.

[38] Consumer Privacy Bill of Rights – The White House. http://www.whitehouse.gov/sites/default/files/privacy-final.pdf.