malware is downloaded by surfing a malicious web site instead of using apps. It is also different since it does not engage in any of the above activities. It may be a reconnaissance tool to reach mobile devices (which may be attached to corporate networks) and possibly try to use them as a jumping board for an attack.

One of the newcomers among Android malware in April 2012 was TigerBot. This malware allows remote access by the attacker and can be controlled via SMS messages [18]. It will listen for specific messages, which can steal contacts lists and screenshots, change network settings, deactivate other software, and control running processes. During the static analysis in this paper, the apps containing this malware have been shown to obtain (if installed) an unusual variety of intrusive permissions. TigerBot is beyond adware, actually a Trojan going so far as to disguise itself with a Google icon.

The Honeynet online community has been at the forefront in showcasing new dynamic analysis tools for Android malware [19]. These tools include DroidBox and APKInspector, which have been trialed during the course of writing this paper. In some ways, these tools do not completely automate all human analysis. Rather, they synergistically integrate individual tools into a suite, run them together with less effort, and produce visual graphs, and work flows that greatly assist the examination of malware. Technically, these tools can be used broadly to help analyze any Android software. However, there are features that make them especially geared toward catching malicious activity, for example screen tabs for permissions and phone calls, and measuring SMS, cryptographic activity, and data leaks.

A honeypot is a computer system that can be used as a trap and be monitored in order to detect and study new attacks. This type of system can help automate security analysis by reducing the manual search for malicious apps and servers, and by capturing live information and audit logging the attacks. HoneyDroid started in 2011 as the first effort to build a honeypot system for the Android platform [20]. Although honeypot ideas were explored in earlier years with Windows Mobile and Symbian, the projects have not matured or stayed on as a result of those platforms being phased out or revamped. The first challenge is making the Android honeypot visible to attackers. Perhaps a main reason that undermines the feasibility of a honeypot system is that smartphones rarely run network services to be noticed by itself on the internet and an active seeking of malicious sites or people would somewhat conflict with the idea of a honeypot itself. The second challenge is that the core components of the honeypot system must themselves not be vulnerable as that would cause them to not function properly for detecting and recording malicious behavior. HoneyDroid's solution to this dilemma is to virtually run Android on top of another securely isolated microkernel operating system, all residing on real phone hardware [20]. The drawback of this approach is that virtualization may be noticed by malware or it may not run the same way in the honeypot.

IV. FINANCIAL MOTIVES

It is not just the core technical attributes of an operating system that determines its exposure to risk; the greater popularity of any operating system platform will lead to more attempts by perpetrators to target that platform and its user base. Nevertheless, obscurity does not mean better security; in comparison to proprietary operating systems, open-source operating systems allow and encourage a greater number of people to work against malware. The body of knowledge suggests that hackers typically do not go about finding vulnerabilities by reading the underlying source code; they do so by probing and trying different tactics from the outside [5]. As a result, brute force attacks are often used. Mike Calce is a famous former hacker, and currently a consultant and the author of a book on internet security. In an interview in 2012, he also stated his belief that the ulterior motive for most of today's hackers is monetary gain [7]. In addition, the risk is shifting more and more from governments and companies to individuals.

Adware is any software package that automatically presents advertisements to users by guessing from their previous surfing or search activities. This involves collecting information, often by user consent, but in some cases, stealing important personal information for ulterior harmful motives. Aside from adware used intentionally by an ad network, other intrusive adware may also exploit an ad network and subvert revenue and information from the owners of the ad network [3]. Social engineering is the art of manipulating people using trickery or deception for the purpose of information gathering, fraud, or system access. Phishing is a common type of social engineering where the attacker notifies users that they need to take action. The email or pop-up contains a link to a fake web site for collecting the user's id and password. Adware may also be used in combination with phishing or automated click fraud.

Some of the pushed ads are displayed through the Android notification bar. The developers of an app can earn part of the advertising revenues this way. This is usually done via cloud messaging, where the server can send notifications to a smartphone without the device requesting them first [6]. This of course requires that the app has gained the necessary permissions to auto-start at boot and run in the background. This may cause drain of resources such as the battery, temporary files and images occupying the device's storage space, and increased internet data usage and roaming charges [6]. Furthermore, if the notifications are clicked on, they may lead to phishing, hacking, or other high risk web sites. Google has updated Google Play Developer Program Policies in August 2012 to prohibit inappropriate advertising activities. With these policies, aside from generally illegal and offensive content, Google is also banning deceptive adware behavior in apps such as impersonating the operating system, making changes to the user's device, hiding from the user which app is generating the ads, and not giving the user the ability to adjust advertising preferences [1].

The analysis of malware and recommendations against them are not based only on program logic because there are supplementary geographic and financial trends. Malware and adware can be better understood in the context of their monetary objectives and countries where they often originate and are distributed (in particular China and Russia). For example, one of the most recent and sophisticated malware that downloads paid apps and media files, leading to unwanted monetary charges, is based in China. The malware called

TROJMMARKETPLAY (discovered by security vendors) comes in multiple versions, some of which even includes experimental code. It changes the smartphone's access point name, connects automatically to a secondary app store, closes normal consent windows, and intercepts verification codes messages so that the user remains unaware in the interim [9].

Secondary (unofficial) app markets, where many malware have been found, seem to have grown also as a result of language factors in the case of China, Hong Kong, and other parts of East Asia. The official Android Market (Google Play) is still blocked as a result of government restrictions in China. This has given rise to many secondary local app markets there.

On one hand, many free apps rely on advertising to support their development. On the other hand, as can be seen from the sample malware, certain apps have crossed the line from merely displaying ads to pushing (or forcing) products to the user, harvesting private data for future use (e.g. spam or other use), and even extracting fraudulent revenues. It is possible for hackers to rent premium rate numbers anonymously (for generating dialing or SMS fraud) in Russia and other Eastern European countries whereas this is not possible in many other countries [17]. This type of fraud affecting Russian Android users goes back to 2010, involved sending SMS to certain numbers that cost the users US$ 5 per message [11]. A recent example of the same SMS fraud activity involved fake Skype apps that were downloaded through Russian web sites as Java MIDlets, which again cause monetary damage [8].

Recent types of Android malware resemble their desktop-based predecessors rather than being genuinely created for a specific operating system. Therefore it is necessary to recall the key motives of the hacker subculture in general that also pertain to malicious Android activity. These are entertainment, ego, status, entrance to a social group, money, and cause [4]. Money, a less common motivator in the 1980s, has grown as a result of the World Wide Web, the enormous volume of commercial transactions, and the vast amount of personal information available and exchanged online [4]. The stolen information (credit cards, bank accounts, logins, etc.) is sold between hackers worldwide. Malware and botnets (collection of compromised computers) are also traded in this underground economy. This black market allows skilled hackers to make a profit by selling their expertise and spoils to others [4].

V. CONCLUSION

Android is presumably the most popular mobile operating system in most countries (including high-income countries). Android has achieved the market breakthrough that the proponents of open-source and Linux software have been waiting for, in economically developing countries as well [12]. Android has been successfully adopted by many hardware manufacturers, with a wide range of expensive and low priced models. By 2015, low-end Android smartphones are expected by market researchers to seize 80% of the market in Africa, India, and China [16]. This great market share across the world also makes Android vulnerable as it provides a large financial incentive for hackers and malware perpetrators to target its individual users and their private information.

Android security will be a crucial area of research for IT security professionals and their academic counterparts. The upside of the current situation is that malware is being quickly disclosed, thanks to accessible and open-source software development tools. Open source software facilitates worldwide community response to security threats. Cooperation against malware needs to increase, not just within individual countries, but across different geographic regions of the world.

In the future, honeypots and dynamic analysis tool suites should put more emphasis on detecting and understanding malware behavior that may have monetary consequences. In order for honeypots to become more visible and successful, they should be deployed in different parts of the world and be capable of operating in different languages. The future holds promise for interesting developments in smartphone security.

REFERENCES

[1] Google (2012). Google Play Developer Program Policies. http://play.google.com/about/developer-content-policy.html (Access Date: 12 September, 2012).

[2] Whitwam, R. (2012). Circumventing Google's Bouncer, Android's anti-malware system. http://www.extremetech.com/computing/130424-circumventing-googles-bouncer-androids-anti-malware-system (Access Date: 29 August, 2012).

[3] Laboratory for Communications and Applications (2011). IEEE COMSOC MMTC E-Letter: ISPs and Ad Networks against Botnet Ad Fraud. http://infoscience.epfl.ch/record/165675/files/E-Letter-Vratonjic.pdf (Access Date: 22 April, 2012) .

[4] Holt, T. and Kilger, M. (2012). Know Your Enemy: The Social Dynamics of Hacking. http://www.honeynet.org/papers/socialdynamics (Access Date: 1 September, 2012).

[5] Mohan, R. (2010). In defense of BIND: open source DNS software yields a better breed of secure product. http://www.securityweek.com/defense-bind-open-source-dns-software-yields-better-breed-secure-product (Access Date: 22 April, 2012).

[6] Hong Kong Computer Emergency Response Team Coordination Centre (2012). Risk Implications of Push Advertisement in Android System. https://www.hkcert.org/my_url/en/blog/12082201 (Access Date: 12 September, 2012) .

[7] Rachal, P. (2012). Former Hacker: Today's Hacks Are All About the Money. http://mashable.com/2012/08/15/mafiaboy-on-hackers/ (Access Date: 30 August, 2012).

[8] Protalinski, E. (2012). Microsoft also warns of fake Skype malware app on Android. http://www.zdnet.com/microsoft-also-warns-of-fake-skype-malware-app-on-android-7000001175/ (Access Date: 1 September, 2012).

[9] Sun, W. (2012). Android Malware Family Downloads Paid Media and Apps. http://blog.trendmicro.com/android-malware-family-downloads-paid-media-and-apps/ (Access Date: 23 August, 2012).

[10] Apvrille, A. (2011). Cryptography for mobile malware obfuscation. RSA Conference Europe, October 2011

[11] Castillo, C. (2011). Android Malware Past, Present, and Future [White Paper]. http://www.mcafee.com/us/resources/white-papers/wp-android-malware-past-present-future.pdf (Access Date: 22 April, 2012).

[12] Erturk, E. (2009). International technology transfer: the case of free computer software. Proceedings of the International Academy of Business and Public Administration Disciplines (IABPAD) 2009 Winter Conference in Orlando, Florida.

[13] Kingsley-Hughes, A. (2012). Android malware uses server-side polymorphism to evade detection. http://www.zdnet.com/blog/hardware/android-malware-uses-server-side-polymorphism-to-evade-detection/17945 (Access Date: 22 April, 2012) .

[14] Los, R. (2012). Vulnerable Open-Source Code in the Enterprise. http://h30499.www3.hp.com/t5/blogs/mobileblogarticlepage/blog-id/sws-119/article-id/884 (Access Date: 22 April, 2012).

[15] Bahwani, C. (2012). Remove Android:Plankton [PUP] Virus from Android Device after Downloading Apps from GetJar. http://androidadvices.com/remove-androidplanktona-pup-virus-android-device-downloading-apps-getjar/#.UEHfC9ZlRt1 (Access Date: 29 August, 2012).

[16] NPD In-Stat (2012). Low-Cost Android Smartphones Will Seize 80% of Market in Africa, India, and China. http://www.instat.com/newmk.asp?ID=3343 (Access Date: 24 May, 2012).

[17] Schwartz, M. (2012). New Android Malware Has Costly Twist. http://www.informationweek.com/news/security/mobile/232600313 (Access Date: 24 May, 2012).

[18] Symantec (2012). Android.Tigerbot Technical Details. http://www.symantec.com/security_response/writeup.jsp?docid=2012-041010-2221-99&tabid=2 (Access Date: 24 May, 2012).

[19] Honeynet Project (2012). To learn the tools, tactics and motives involved in computer and network attacks, and share the lessons learned. http://www.honeynet.org/blog (Access Date: 28 August, 2012).

[20] C. Mulliner, S. Liebergeld, and M. Lange (2011), Poster: Honeydroid-creating a smartphone honeypot. IEEE Symposium on Security and Privacy, May 2011.

