International Journal for Modern Trends in Science and Technology, 8(11): 133-139, 2022

ISSN: 2455-3778 online
DOI: https://doi.org/10.46501/IJMTST0811023
Available online at: http://www.ijmtst.com/vol8issue11.html

Android Malware Forensic Investigation Using Machine Learning

Anand Kumar Srivastava

Research Scholar, Dept. of Computer Science, Sharda University, Greater Noida, India

To Cite this Article
Anand Kumar Srivastava. Android Malware Forensic Investigation Using Machine Learning. International Journal for Modern Trends in Science and Technology 2022, 8(11), pp. 133-139. https://doi.org/10.46501/IJMTST0811023

Article Info
Received: 18 October 2022; Accepted: 05 November 2022; Published: 17 November 2022.

ABSTRACT

Android applications have recently witnessed a pronounced progress, making them among the fastest growing technological fields to thrive and advance. However, such level of growth does not evolve without some cost. This particularly involves increased security threats that the underlying applications and their users usually fall prey to. As malware becomes increasingly more capable of penetrating these applications and exploiting them in suspicious actions, the need for active research endeavors to counter these malicious programs becomes imminent. Some of the studies are based on dynamic analysis, and others are based on static analysis, while some are completely dependent on both. This study is static, dynamic, and hybrid analyses to identify malicious applications. We leverage machine learning classifiers to detect malware activities as we explain the effectiveness of these classifiers in the classification process. The results prove the efficiency of permissions and the action repetition feature set and their influential roles in detecting malware in Android applications. The results show empirically very close accuracy results when using static, dynamic, and hybrid analyses. Thus, using static analyses due to their lower cost compared to dynamic and hybrid analyses. In other words, there are best results in terms of accuracy and cost (the trade-off) make us select static analysis over other techniques.

Keywords: android, forensic, machine, investigation, learning, analysis, dynamic, applications, efficiency

1. INTRODUCTION

Ransomware attacks are not only limited to Personal Computers but are increasing rapidly to target smart-phones as well. The attackers target smart-phone devices to steal users' personal information for monetary purposes. However, Android is the most widely used mobile operating system with the largest market share in the world that makes it a primary target for cyber-criminals to attack.[1,2] The existing research towards the detection of Android ransomware lacks significant features and works with supervised machine learning techniques. But there are several restrictions in supervised machine learning techniques such as these techniques heavily rely on anti-virus vendors to provide explicit labels and the given sample can be wrongly classified if the training set does not include related examples and/or if the labels are incorrect. Moreover, it may not detect unknown ransomware samples in real-time situations due to the absence of historical targets in the real world. In a study, an attempt is made for an in-depth investigation of Android ransomware with reverse engineering and forensic analysis to extract static features.[3,4] Furthermore, a novel RansomDroid framework on clustering based unsupervised machine

learning techniques is proposed to address the issues such as mislabeling of historical targets and detecting unforeseen Android ransomware. To the best of our knowledge, performing unsupervised machine learning techniques for the detection of Android ransomware is still an open area of research that has not been explored by the researchers yet. The proposed RansomDroid framework employs a Gaussian Mixture Model that has a flexible and probabilistic approach to model the dataset. RansomDroid framework utilizes feature selection and dimensionality reduction to further improve the performance of the model. The prior study results show that the proposed RansomDroid framework detects Android ransomware with an accuracy of 98.08% in 44 ms.[5,6]

Android is the most popular operating system among mobile devices and the malware targeted explicitly for Android is rapidly growing and spreading across the mobile ecosystem. In another paper, we proposed a hybrid analysis of Android malware to retrieve evidential data, generated from or accessed by such mobile malware, which can be adopted as critical evidence for civil and criminal cases. We targeted on Android malware from Joker Family where we collected and analyzed 62 recently discovered malicious apps, we found that: 11 apps access and store user's location information, 17 apps track user's SMS text messages and 58 apps send out user personal information to remote servers. Our proposed approach found that, evidence data including location, timestamp, IP address are still able to be identified from the local file system and logging system. Our main contribution in research was to provide an effective forensic analysis report on Android malware that can extract critical evidence from the local file systems as well as system logs.[7,8]

Android has progressively grown to become in a few years the most widely used smartphone operating system . With more and more users relying on Android-enabled handheld device, and able to install third party applications from official and alternative markets, the security of both devices and the underlying network becomes an essential concern for both the end user and his service provider. In recent years, practitioners and researchers have witnessed the emergence of a variety of Android malware. [9,10] The associated threats range from simple user tracking and disclosure of personal information to advanced fraud and premium-rate SMS services subscription, or even unwarranted involvement in botnets. Although most users are nowadays aware that personal computers can and will be attacked by malware, very few realize that their smartphone is prone to an equally dangerous threat. To assess the threat of software downloaded from the internet, discerning users rely on scan results yielded by antivirus products. Unfortunately, each antivirus vendor has its secret recipe on how/why it decides to assign a malware label to a given application. Thus, an application can be differently appreciated by distinct antivirus products, leading to damaging confusions. Indeed, both practitioners and researchers heavily rely on antivirus, whether to trust apps or to build the ground truths for assessment tasks.[11,12]

Crimes do not happen in isolation from technological tendencies; therefore, mobile device forensics has become a significant part of digital forensics.[13,14]

Most people do not realize how complicated the mobile forensics process can be in reality. As the mobile devices increasingly continue to gravitate between professional and personal use, the streams of data pouring into them will continue to grow exponentially as well. Did you know that 33,500 reams of paper are the equivalent of 64 gigabytes if printed? Storage capacity of 64 GB is common for today's smartphones.

The mobile forensics process aims to recover digital evidence or relevant data from a mobile device in a way that will preserve the evidence in a forensically sound condition. To achieve that, the mobile forensic process needs to set out precise rules that will seize, isolate, transport, store for analysis and proof digital evidence safely originating from mobile devices.[15,16]

Usually, the mobile forensics process is similar to the ones in other branches of digital forensics. Nevertheless, one should know that the mobile forensics process has its own particularities that need to be considered. Following correct methodology and guidelines is a vital precondition for the examination of mobile devices to yield good results.

Among the figures most likely to be entrusted with the performance of the following tasks are Forensic

Examiners, Incident Responders, and Corporate Investigators. During the inquiry into a given crime involving mobile technology, the individuals in charge of the mobile forensic process need to acquire every piece of information that may help them later – for instance, device's passwords, pattern locks or PIN codes.[17,18]

2. DISCUSSION

Digital forensics operates on the principle that evidence should always be adequately preserved, processed, and admissible in a court of law. Some legal considerations go hand in hand with the confiscation of mobile devices.

There are two major risks concerning this phase of the mobile forensic process: Lock activation (by user/suspect/inadvertent third party) and Network / Cellular connection.

Network isolation is always advisable, and it could be achieved either through 1) Airplane Mode + Disabling Wi-Fi and Hotspots, or 2) Cloning the device SIM card.[19,20]

Mobile devices are often seized switched on; and since the purpose of their confiscation is to preserve evidence, the best way to transport them is to attempt to keep them turned on to avoid a shutdown, which would inevitably alter files.

A Faraday box/bag and external power supply are common types of equipment for conducting mobile forensics. While the former is a container specifically designed to isolate mobile devices from network communications and, at the same time, help with the safe transportation of evidence to the laboratory, the latter, is a power source embedded inside the Faraday box/bag. Before putting the phone in the Faraday bag, disconnect it from the network, disable all network connections (Wi-Fi, GPS, Hotspots, etc.), and activate the flight mode to protect the integrity of the evidence. Last but not least, investigators should beware of mobile devices being connected to unknown incendiary devices, as well as any other booby trap set up to cause bodily harm or death to anyone at the crime scene.[21,22]

The goal of this phase is to retrieve data from the mobile device. A locked screen can be unlocked with the right PIN, password, pattern, or biometrics (Note that biometric approaches while convenient are not always protected by the fifth amendment of the U.S. Constitution). According to a ruling by the Virginia Circuit Court, passcodes are protected, fingerprints not. Also, similar lock measures may exist on apps, images, SMSs, or messengers. Encryption, on the other hand, provides security on a software and/or hardware level that is often impossible to circumvent.

It is hard to be in control of data on mobile devices because the data is mobile as well. Once communications or files are sent from a smartphone, control is lost. Although there are different devices having the capability to store considerable amounts of data, the data in itself may physically be in another location. To give an example, data synchronization among devices and applications can take place directly but also via the cloud. Services such as Apple's iCloud and Microsoft's One Drive are prevalent among mobile device users, which leave open the possibility for data acquisition from there. For that reason, investigators should be attentive to any indications that data may transcend the mobile device as a physical object, because such an occurrence may affect the collection and even preservation process.[23,24]

Since data is constantly being synchronized, hardware and software may be able to bridge the data gap. Consider Uber – it has both an app and a fully functional website. All the information that can be accessed through the Uber app on a phone may be pulled off the Uber website instead, or even the Uber software program installed on a computer.

Regardless of the type of the device, identifying the location of the data can be further impeded due to the fragmentation of operating systems and item specifications. The open-source Android operating system alone comes in several different versions, and even Apple's iOS may vary from version to version.

Another challenge that forensic experts need to overcome is the abundant and ever-changing landscape of mobile apps. Create a full list of all installed apps. Some apps archive and backup data.[25,26]

After one identifies the data sources, the next step is to collect the information properly. There are certain unique challenges concerning gathering information in the context of mobile technology. Many mobile devices cannot be collected by creating an image and instead they may have to undergo a process called acquisition of data. Thera are various protocols for collecting data from mobile devices as certain design specifications may only allow one type of acquisition.

The forensic examiner should make a use of SIM Card imagining – a procedure that recreates a replica image of the SIM Card content. As with other replicas, the original evidence will remain intact while the replica image is being used for analysis. All image files should be hashed to ensure data remains accurate and unchanged.[27,28]

3. RESULTS

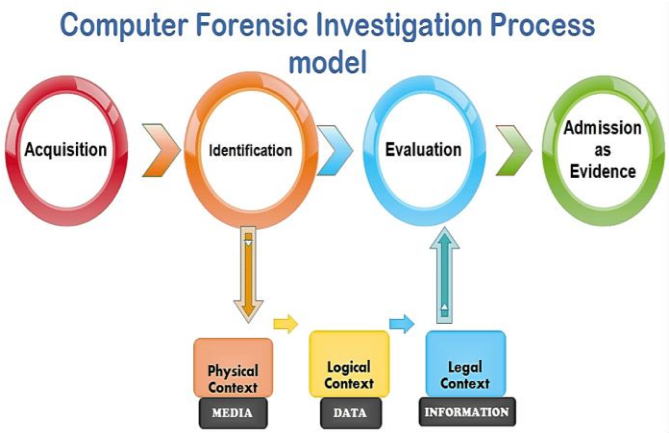
As the first step of every digital investigation involving a mobile device(s), the forensic expert needs to identify:

- Type of the mobile device(s) – e.g., GPS, smartphone, tablet, etc.
- Type of network – GSM, CDMA, and TDMA
- Carrier
- Service provider (Reverse Lookup)[29,30]

The examiner may need to use numerous forensic tools to acquire and analyze data residing in the machine. Due to the sheer diversity of mobile devices, there is no one-size-fits-all solution regarding mobile forensic tools. Consequently, it is advisable to use more than one tool for examination. AccessData, Sleuthkit, and EnCase are some popular forensic software products that have analytic capabilities. The most appropriate tool(s) is being chosen depending on the type and model of mobile device.

Timeline and link analysis available in many mobile forensic tools could tie each of the most significant events, from a forensic analyst's point of view.

All of the information, evidence, and other findings extracted, analyzed, and documented throughout the investigation should be presented to any other forensic examiner or a court in a clear, concise, and complete manner.[31,32]



No matter what your actual mobile forensic method is, it is imperative to create a policy or plan for its execution and follow all its steps meticulously and in the proper sequence. Not following the protocol may entail grave consequences. One should start with non-invasive forensic techniques first as they tend to endanger a device's integrity to a lesser degree. Be careful with built-in security features – "[f]or example, collecting a physical image before a logical image on certain devices can completely wipe a phone of all data, as can attempting to access a locked device and making too many password attempts."

From the legal point of view, the level of the interaction between the user and the device is critical.[33,34]

Non-invasive methods can deal with other tasks, such as unlocking the SIM lock or/and the operator lock, the operating system update, IMEI number modification, etc. These techniques are virtually inapplicable in cases where the device has sustained severe physical damage. Types of non-invasive mobile forensic methods:

- Manual extraction
  The forensic examiner merely browses through the data using the mobile device's touchscreen or keypad. Information of interest discovered on the phone is photographically documented. This process of manual extraction is simple and applicable to almost every phone. While there are some tools designed to make this process easier, it is not possible, however, to restore deleted data this way.[35]

- Logical extraction

This approach involves instituting a connection between the mobile device and the forensic workstation using a USB cable, Bluetooth, Infrared or RJ-45 cable. Following the connecting part, the computer sends command requests to the device, and the device sends back data from its memory. The majority of forensic tools support logical extraction, and the process itself requires short-term training. On the downside, however, this technique may add data to the mobile device and may alter the integrity of the evidence. Also, deleted data is rarely accessible.

- JTAG method

  JTAG is a non-invasive form of physical acquisition that could extract data from a mobile device even when data was difficult to access through software avenues because the device is damaged, locked or encrypted. The device, however, must be at least partially functional (minor damages would not hinder this method).

  The process involves connecting to the Test Access Ports (TAPs) on a device and instructing the processor to transfer raw data stored on connected memory chips. This is a standard feature that one could come across in many mobile phone models, which provides mobile phone manufactures a low-level interface outside the operating system. Digital forensic investigators take an interest in JTAG, as it can, in theory, allow direct access to the mobile device's memory without jeopardizing it. Despite that fact, it is a labor-intensive, time-consuming procedure, and it requires advance knowledge (not only of JTAG for the model of the phone under investigation but also of how to arrange anew the resulting binary composed of the phone's memory structures).[36]

- Hex dump

  Similar to JTAG, Hex dump is another method for physical extraction of raw information stored in flash memory. It is performed by connecting the forensic workstation to the device and then tunneling an unsigned code or a bootloader into the device, each of them will carry instructions to dump memory from the phone to the computer. Resulting image is fairly technical—in binary format—and it requires a person having the technical education to analyze it. Furthermore, the examiner comes into possession of an abundant amount of data, since deleted data can be recovered, and, on top of that, the entire process is inexpensive.

  Typically, they are longer and more complex. In cases where the device is entirely non-functional due to some severe damage, it is very likely the only way to retrieve data from the device might be to manually remove and image the flash memory chips of the device. Even if the device or item is in good condition, circumstances may require the forensic expert to acquire the chip's contents physically.[37]

- Chip-off

  A process that refers to obtaining data straight from the mobile device's memory chip. According to the preparations pertinent to this level, the chip is detached from the device and a chip reader or a second phone is used to extract data stored on the device under investigation. It should be noted that this method is technically challenging because of the wide variety of chip types existing on the mobile market. Also, the chip-off process is expensive, training is required, and the examiner should procure specific hardware to conduct de-soldering and heating of the memory chip. Bits and bytes of raw information that is retrieved from the memory are yet to be parsed, decoded, and interpreted. Even the smallest mistake may lead to damages to the memory chip, which, in effect, would render the data irrevocably lost. Consequently, experts advise having recourse to chip-off when: a) other methods of extraction are already attempted, b) it is important to preserve the current state of device's memory, c) the memory chip is the only element in a mobile device that is not broken.[38]

  The whole process consists of five stages:

1. Detect the memory chip typology of the device
2. Physical extraction of the chip (for example, by unwelding it)
3. Interfacing of the chip using reading/programming software

4. Reading and transferring data from the chip to a PC
5. Interpretation of the acquired data (using reverse engineering)

The last two phases coincide with those of the non-invasive methods. However, the phases of physical extraction and interfacing are critical to the outcome of the invasive analysis.

- Micro read

  This method refers to manually taking an all-around view through the lenses of an electron microscope and analyzing data seen on the memory chip, more specifically the physical gates on the chip. In a nutshell, micro read is a method that demands utmost level of expertise, it is costly and time-consuming, and is reserved for serious national security crises.[38,39]

4. CONCLUSIONS

MOBILedit Forensic is an all-in-one solution for data extraction from phones, smartwatches and clouds. It utilizes both physical and logical data acquisition, has excellent application analysis, deleted data recovery, a wide range of supported devices, fine-tuned reports, concurrent processing, and easy-to-use interface. With a brand new approach, MOBILedit Forensic is much stronger in security bypassing than ever before.

MOBILedit Forensic offers maximum functionality at a fraction of the price of other tools. It can be used as the only tool in a lab or as an enhancement to other tools with its data compatibility. When integrated with Camera Ballistics it scientifically analyzes camera photo origins. With MOBILedit Forensic, you can extract all the data from a phone with only a few clicks. This includes deleted data, call history, contacts, text messages, multimedia messages, photos, videos, recordings, calendar items, reminders, notes, data files, passwords, and data from apps such as Skype, Dropbox, Evernote, Facebook, WhatsApp, Viber, Signal, WeChat and many others.

MOBILedit Forensic automatically uses multiple communication protocols and advanced techniques to get maximum data from each phone and operating system. Then it combines all data found, removes any duplicates and presents it all in a complete, easily readable report.[40]

Conflict of interest statement

Authors declare that they do not have any conflict of interest.

REFERENCES

[1] Aafer, Du & Yin (2013) Aafer Y, Du W, Yin H. DroidAPIMiner: mining API-Level Features for Robust Malware Detection in Android. Security and privacy in communication networks; Sydney, NSW, Australia. 2013. pp. 86–103. [Google Scholar]

[2] Abadi, Agarwal & Barham (2016) Abadi M, Agarwal A, Barham P. TensorFlow: large-scale machine learning on heterogeneous distributed systems. 2016. http://arxiv.org/abs/1603.04467.

[3] Aktas & Sen (2018) Aktas K, Sen S. UpDroid: updated Android Malware and its familial classification. In: Gruschka N, editor. Secure IT Systems. NordSec 2018. Lecture Notes in Computer Science. Vol. 11252. Springer; Cham: 2018. [CrossRef] [Google Scholar]

[4] Al-Rfou et al. (2016) Al-Rfou R, Alain G, Almahairi A, Angermueller C, Bahdanau D. Theano: A Python framework for fast computation of mathematical expressions. arXiv. 2016. . http://arxiv.org/abs/1605.02688.

[5] Allahham & Rahman (2018) Allahham AA, Rahman MA. A smart monitoring system for campus using zigbee wireless sensor networks. International Journal of Software Engineering & Computer System. 2018;4(1):1–14. doi: 10.15282/ijsecs.4.1.2018.1.0034. [CrossRef] [Google Scholar]

[6] Allix et al. (2018) Allix K, Bissyandé TF, Klein J, Le Traon Y. AndroZoo: collecting millions of android apps for the research community. MSR '16 proceedings of the 13th international conference on mining software repositories, Austin, Texas; 2018. pp. 468–471. [CrossRef] [Google Scholar]

[7] Alotaibi (2019) Alotaibi A. Identifying malicious software using deep residual long-short term memory. IEEE Access. 2019;7:163128–163137. doi: 10.1109/ACCESS.2019.2951751. [CrossRef] [Google Scholar]

[8] Alsoghyer & Almomani (2019) Alsoghyer S, Almomani I. Ransomware detection system for android applications. Electron. 2019;8(8):1–36. doi: 10.3390/electronics8080868. [CrossRef] [Google Scholar]

[9] Alswaina & Elleithy (2018) Alswaina F, Elleithy K. Android malware permission-based multi-class classification using extremely randomized trees. IEEE Access. 2018;6:76217–76227. doi: 10.1109/ACCESS.2018.2883975. [CrossRef] [Google Scholar]

[10] Alswaina & Elleithy (2020) Alswaina F, Elleithy K. Android malware family classification and analysis: current status and future directions. Electron. 2020;9(6):1–20. doi: 10.3390/electronics9060942. [CrossRef] [Google Scholar]

[11] Amit & Geman (1997) Amit Y, Geman D. Shape quantization and recognition with randomized trees. Neural Computation. 1997;9(7):1545–1588. doi: 10.1162/neco.1997.9.7.1545. [CrossRef] [Google Scholar]

[12] Anderson, Filar & Roth (2017) Anderson HS, Filar B, Roth P. BlackHat DC. 2017. Evading Machine Learning Malware Detection; p. 6. . https://www.blackhat.com/docs/us-17/thursday/us-17-Anderson-Bot-Vs-Bot-Evading-Machine-Learning-Malware-Detection-wp.pdf. [Google Scholar]

[13] Anderson et al. (2018) Anderson HS, Kharkar A, Filar B, Evans D, Roth P. Learning to Evade Static PE Machine Learning Malware Models via Reinforcement Learning. ArXiv. 2018. . http://arxiv.org/abs/1801.08917.

[14] Android Developers (2017) Android Developers . Google Developers; 2017. Android Debug Bridge (ADB) [Google Scholar]

[15] Android (2013) Android aapt. 2013. https://androidaapt.com/

[16] Android (2019a) Android What is Android. 2019a. [30 September 2019]. https://www.android.com/what-is-android/
[17] Android (2019b) Android ProGuard. 2019b. [02 October 2019]. https://developer.android.com/studio/build/shrink-code
[18] Android Developer (2020) Android Developer . 2020. Opcodes. Google Developers. [Google Scholar]
[19] Android Developer (2018) Android Developer Dedexer. 2018. [02 October 2019]. http://dedexer.sourceforge.net/
[20] Android (2015) Android App Manifest. 2015. [02 October 2019]. https://developer.android.com/guide/topics/manifest/manifest-intro
[21] Anonymous (2019b) Anonymous. Baksmali & Smali. 2019b. [02 October 2019]. https://github.com/JesusFreke/smali/wiki
[22] Apvrille & Strazzere (2012) Apvrille A, Strazzere T. Reducing the window of opportunity for Android malware Gotta catch 'em all. Journal of Computer Virology. 2012;8(1–2):61–71. doi: 10.1007/s11416-012-0162-3. [CrossRef] [Google Scholar]
[23] Apvrille & Apvrille (2013) Apvrille L, Apvrille A. Pre-filtering Mobile Malware with Heuristic Techniques. The 2nd international symposium on research in grey-hat hacking (GreHack); Grenoble, France. 2013. pp. 1–16. [Google Scholar]
[24] Aresu et al. (2016) Aresu M, Ariu D, Ahmadi M, Maiorca D, Giacinto G. Clustering android malware families by http traffic. 2015 10th Int. conf. malicious unwanted software, MALWARE 2015; 2016. pp. 128–135. [CrossRef] [Google Scholar]
[25] Arp et al. (2014) Arp D, Spreitzenbarth M, Malte H, Gascon H, Rieck K. DREBIN: effective and explainable detection of android malware in your pocket. 21th annual network and distributed system security symposium (NDSS); San Diego, CA. 2014. pp. 1–15. [Google Scholar]
[26] ASUS (2019a) ASUS Asus. 2019a. [02 October 2019]. https://www.asus.com/my/
[27] Atici, Sagiroglu & Dogru (2016) Atici MA, Sagiroglu S, Dogru IA. Android malware analysis approach based on control flow graphs and machine learning algorithms. IEEE 4th international symposium on digital iforensics and security, little rock, Arkansas; Piscataway. 2016. pp. 26–31. [CrossRef] [Google Scholar]
[28] Atzeni et al. (2018) Atzeni A, Diaz F, Marcelli A, Sanchez A, Squillero G, Tonda A. Countering android malware: A scalable semi-supervised approach for family-signature generation. IEEE Access. 2018;6:59540–59556. doi: 10.1109/ACCESS.2018.2874502. [CrossRef] [Google Scholar]
[29] Aubrey-Derrick Schmidt et al. (2009a) Aubrey-Derrick Schmidt OK, Bye R, Schmidt H-G, Clausen J, Kamer SA, Yuksel A, Camtepe SA. Static Analysis of Executables for Collaborative Malware Detection on Android. IEEE international conference on communications (ICC). Dresden, Germany; Piscataway. 2009a. pp. 1–5. [CrossRef] [Google Scholar]
[30] Aubrey-Derrick Schmidt et al. (2009b) Aubrey-Derrick Schmidt SA, Schmidt H-G, Batyuk L, Clausen J, Camtepe SA. Smartphone malware evolution revisited: android next target?. 2009, 4th int. conf. malicious unwanted software, MALWARE 2009; 2009b. pp. 1–7. [CrossRef] [Google Scholar]
[31] Aung & Zaw (2013) Aung Z, Zaw W. Permission-based android malware detection. International Journal of Scientific and Technology Research. 2013;2(3):228–234. [Google Scholar]
[32] Badhani & Muttoo (2019) Badhani S, Muttoo SK. CENDroid—A cluster-ensemble classifier for detecting malicious Android applications. Computers & Security. 2019;85:25–40. doi: 10.1016/j.cose.2019.04.004. [CrossRef] [Google Scholar]
[33] Bartel et al. (2012) Bartel A, Klein J, Le Traon Y, Monperrus M. Automatically securing permission-based software by reducing the attack surface: an application to Android. Proceedings of the 27th IEEE/ACM international conference on automated software engineering (ASE). Essen, Germany; Piscataway. 2012. pp. 274–277. [CrossRef] [Google Scholar]
[34] Battista et al. (2016) Battista P, Mercaldo F, Nardone V, Santone A, Visaggio CA. Identification of android malware families with model checking. Proceedings of the 2nd international conference on information systems security and privacy (ICISSP); 2016. pp. 542–547. [CrossRef] [Google Scholar]
[35] Beal (2013) Beal V. Mobile malware. 2013. https://www.webopedia.com/TERM/M/mobile_malware.html
[36] Beverly, Garfinkel & Cardwell (2011) Beverly R, Garfinkel S, Cardwell G. Forensic carving of network packets and associated data structures. Digital Investigation. 2011;8:S78–S89. doi: 10.1016/j.diin.2011.05.010. [CrossRef] [Google Scholar]
[37] Blanc et al. (2019) Blanc W, Hashem LG, Elish KO, Hussain Almohri MJ. Identifying Android malware families using Android-oriented metrics. 2019 IEEE International Conference on Big Data (Big Data); Piscataway. 2019. pp. 4708–4713. [CrossRef] [Google Scholar]
[38] Borja Sanz et al. (2013) Borja Sanz GÁ, Santos I, Laorden C, Ugarte-Pedrero X, Nieves J, Bringas PG. MAMA : manifest analysis for malware detection in android. Cybernetics and Systems. 2013 doi: 10.1080/01969722.2013.803889. [CrossRef] [Google Scholar]
[39] Calleja et al. (2018) Calleja A, Martín A, Menéndez HD, Tapiador J, Clark D. Picking on the family: disrupting android malware triage by forcing misclassification. Expert Systems with Applications. 2018;95:113–126. doi: 10.1016/j.eswa.2017.11.032. [CrossRef] [Google Scholar]
[40] Castillo (2011) Castillo CA. Android malware past, present, and future. https://qdoc.tips/android-malware-analysis-pdf-free.html McAfee white paper, Mobile Security Working Group. 2011