MOBILE FORENSIC
(COCS71192)

BY

MUYIDEEN KAZEEM OLUWADARE (21027842L)
COMPUTER SCIENCE (CYBER SECURITY)

17TH January 2023

Table of Contents

Table of Figures

Table of Tables

Part A

1.0 Introduction

Based on recent data, it's clear that mobile devices have surpassed desktop computers as the most prevalent interface for retrieving and accessing information, and the meteoric increase in mobile usage is a boon for researchers in information retrieval and data mining; however, mobile devices can accurately predict users' intent by capturing rich contextual and personal signals, allowing them to serve more pertinent material, and perhaps even offer novel, preemptive, zero-query suggestions, examples are Siri by Apple, Google Now, and Cortana by Microsoft are the examples of such new systems (Wang et al., 2017). Also, unlike their desktop counterparts, sensor data is constantly being left behind by mobile devices (such as GPS or motion sensors), and the user data activity (such as installed applications) with novel sources of implicit, explicit user feedback which allow for the discovery of insightful actionable knowledge and the development of better systems to deliver relevant content to each user at the optimal time, place, and by collecting data on mobile interactions across users which we can draw some fascinating conclusions beyond search, recommendation, and the creation of traffic estimates in real-time is one example of such an application (Wang et al., 2017).

1.1 Mobile Technologies and Features

Despite the proliferation of mobile devices like smartphones, and tablets over the past two decades and the development of numerous new standards for enabling technologies like cellular data networks, Bluetooth, and Wi-Fi, mobile communication was previously limited to character-limited text messages and voice calls over cellular data networks (Venkata Sai & Li, 2020). The Internet became widely used toward the end of the 1990s and had become a widely available commodity, and its growing popularity meant that more and more people were using it to surf the website which is consistent with the findings of Venkata Sai and Li (2020), who note that cellular service providers have begun offering data plans for the internet allowing users to stay connected and surf the web regardless of where they may be. Mobile technology is described as technology that can be brought with the user wherever they go; however, wireless technologies, which include cellular networks, 4G & 5G networks, WIFI, and Bluetooth allow mobile devices to communicate speech, data, and applications (mobile apps) which are ubiquitous, increasing, and this is due to the fact that there are now more than 3 billion smartphone users and an estimated 1.87 billion mobile workers worldwide by 2022 (IBM, 2022).

1.2 Latest Innovation in Mobile Devices and Applications

Since the first telephone was invented in 1870, telephony and numerous materials research have driven this field of technology that creates, implements, and distributes telecommunication services especially recently, and no single event has accelerated telephone adoption, therefore, before World War II, only 62% of American homes had telephones, and this remained true until 1950, where telegrams, letters, and meetings were the main forms of communication, and natural disasters often silenced families, making communication difficult (Moss, 2021). Researchers imagined hybrid telephony and computing devices as early as the 1970s, and about 20 years later in the early 1990s, several prototype multifunction devices became available simultaneously launched in the late '90s, IBM's Simon, Nokia's Communicator, and Qualcomm's PDQ brought together even more computing power and phone capabilities and until 1999, when Japan's NTT DoCoMo introduced the i-mode system, these gadgets were more commonly known as feature phones than smartphones (Islam & Want, 2014). Conforming with Li et al. (2010), smartphones are defined as mobile devices with more powerful computing capabilities and network connections than traditional feature phones which as a rule have the same kinds of cameras, WiFi, app-installing mechanisms, user interface features (like touch screens), and processing power as a regular desktop PC. As mentioned by Toppo & Dhote (2021), the new stage of the development of technology, from desktop PCs to laptops to mobile phones, is more advanced than the previous one, and when it comes to big tech, the battle between iOS and Android represents the classic competitor, however, In December of 2022, the global market share of mobile operating systems was split between iOS (26.98%) and Android (72.37%) (StatCounter, 2023).

1.3 History and Evolution of Mobile Phones.

As reported by Zinkus et al., (2021), the widespread availability of mobile devices has greatly increased the quantity of data that people may take with them at any given moment, however, this has also made people more vulnerable to attacks from those who would like to steal their personal information which has made the manufacturers of both hardware and software respond to these concerns over the past decade by rolling out several significant upgrades to smart device hardware and operating systems, which include the use of passcodes, authentication via biometric by default, as well as the integration of strong encryption mechanisms to secure data at rest and in transit. Smartphones are being utilized more frequently in healthcare settings, either by medical professionals or by patients themselves, for instance, there is a mobile app for documenting wounds and, by extension, enhancing general workflows in clinics through digitalization with the use of tablet devices to help patients designate pain sites using a

sketching module (Schobel et al., 2021). Corresponding to Phongtraychack & Dolgaya (2018), users become increasingly reliant on smartphone apps for problem-solving applications that are widely available in app stores which have a greater chance of turning visits into sales because it streamlines the process of solving problems like booking movie tickets, checking sports scores, buying, and selling, and many similar routine activities that can be solved with a click of the mouse. Common features found on today's smartphones consist of numerous applications and the ability to make and receive calls, play various types of media (such as music or videos), access the internet, use GPS, and view augmented reality (AR) overlays, which have seen a rise in development efforts (Woo et al., 2022).

1.4 Mobile Phone Information for Forensic Investigation

Extraction of data is getting more comprehensive and automated across a wide variety of devices and the ability to selectively extract only the relevant data is also growing in prominence; however, several cases of these recovery work are used to retrieve specific data from Android and iOS mobile devices together with the Android and IOS based smartwatches like the Apple Watch, Samsung Gear 2 Neo, and others (Dorai et al., 2020). As explained by Alatawi et al. (2020), a great deal of information can be gathered on a smartphone itself, additional sources of information may include the user's personal computer with which the device synchronizes and backs up data, as well as the telecommunications provider, especially in cases involving fraud or theft, a forensic investigator can access a variety of data sets related to a device during an investigation, including but not limited to the following:

- GPS data
- Device's call log
- Message history
- Contacts
- Calendar
- Mobile device location
- Audio, and Video Recording
- Social Media Information (WhatsApp, Facebook, Twitter) messages and Logs
- Network Information
- Device Information
- Internet Browsing History
- Memory cards

1.5 Mobile Devices Forensic Analysis, Tools, and Technologies

The analysis phase of a mobile device investigation is the most important and time-consuming part of the process. Due to the size of the storage space and the complexity of the investigation, professional forensic processes and equipment become crucial to analyze the large amounts of data received (Alzaabi, 2013). Tools that facilitate forensic analysis of systems are essential for mobile device forensics investigators, however, the forensics examiner will use a digital image/copy of the device to be analyzed to ensure the integrity of the evidence. (Institute of Electrical and Electronics Engineers, 2022). According to Tajuddin & Manaf (2015), Cellebrite UFED, MOBILedit Forensic, Oxygen, Forensic Toolkit, XRY, EnCASE, JTAG, and Paraben's Device Seizure are popular forensic tools for acquisition and analysis, and they have distinct characteristics and information-gathering abilities on different devices that provide similar services and use similar analysis methods, but they use different report application software to present the data, however, mobile device data extraction can be done using physical and logical acquisition. Direct memory access is used to copy the physical storage device bit-by-bit, while logical acquisition extracts files and directories from a file system (Da Costa et al., 2022). Da Costa et al. (2022), further explained that Logical extraction methods take out all data such as calls, contacts, sms, calendars, photos, etc., while data stored such as deleted data including system & network provider information on a device can be retrieved, created, and otherwise analyzed through the use of physical extraction techniques without SIMs, bypass handset security codes, and analyze memory cards, but it produces Hex data that must be decoded, so investigators benefit from logical extraction.

1.5.1 Mobile Devices Forensic Procedure and Standard

In agreement with Riadi et al. (2018), NIST considers the following steps to be the four pillars of a forensic workflow methodology: collection, examination, analysis, and reporting. The ever-increasing demand for forensic analysis can be directly attributed to the proliferation of smartphones. When a crime is being investigated, forensics experts may need to access and examine data from mobile devices, and to avoid running into the same issues when attempting to glean useful information from a mobile phone, the mobile forensic standard has been developed such as collection, identification, preparation, isolation, processing and verification, documentation and reporting, and finally archiving. (Dovydas Patapas, 2021).

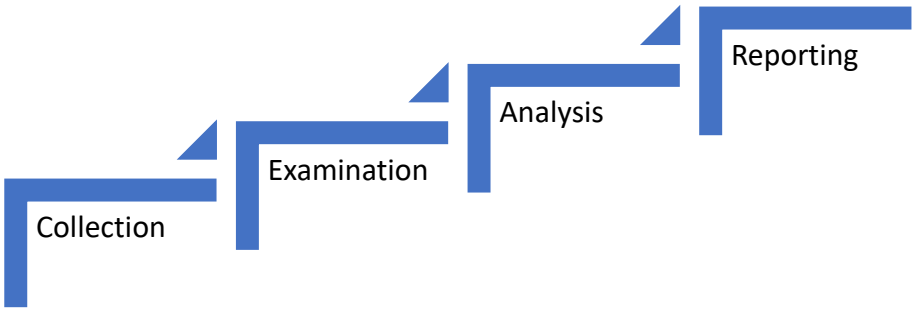


Figure 1 - Forensic Workflow Standard

1.6 Conclusion

This section explains why mobile devices are preferable to desktop computers when it comes to retrieving information because of their streamlined design for end users. By analyzing the user's context and behaviour, mobile devices can provide more pertinent content and make zero-query recommendations. In December of 2022, Android accounted for 72.37 percent of the mobile operating system market, while iOS held 26.98 percent. With so much data necessitating forensic processes and equipment, mobile device analysis is also the most time-consuming and crucial part of the investigation. Due to the rising popularity of smartphones, there is a greater need than ever for forensic analysis, which is where mobile forensics tools come in. During a criminal investigation, forensics experts may examine data from mobile devices. To do so, they must acquire the data, identify the data, prepare the data, isolate the data, verify the data, document the data, and give a report of evidence found from the data acquired.

Part B

2.0 Overview of Android architecture

Since its release in November 2007, Android's intelligent platform has guaranteed the exclusives barrier to the revolution in the mobile device industry by enhancing the functionality of the unified platform system for mobile devices (Chen, 2021). According to Anna University (2022), an open-source Linux Kernel and a collection of C/C++ libraries with application framework APIs are just two parts of the Android architecture that work together to satisfy the needs of any Android device; however, the DVM (Dalvik Virtual Machine) supplies a framework for an Android app to run on top of the Linux Kernel, which is responsible for the main features of the smartphone's OS. In accordance with Rao & Chakravarthy (2016), while Android applications are packaged in the APK setup, the Dalvik VM provides a sandbox environment in which numerous software applications can run simultaneously with a distinct process id, and each process executing in a separate virtual machine (.dex) and the device's internal memory stores the operating system, core libraries, databases, configuration files, apps, data, user data; the SD card stores additional data for application, backups, and other external data for users, however, the Android's architectural stack includes apps, frameworks, libraries, Linux kernel, and the environment used for Android. Figure 2 depicts how each layer's elements are merged for optimal development and execution (Rao & Chakravarthy, 2016).

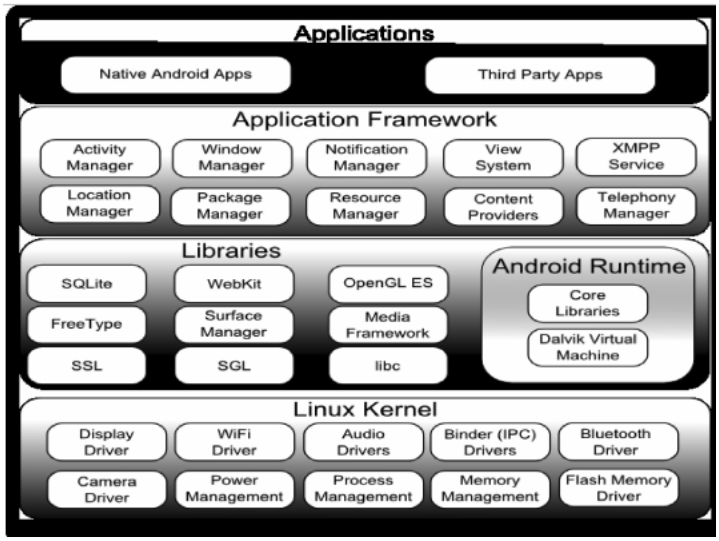


Figure 2 - Android Architecture (Rao & Chakravarthy, 2016)

2.1 Android Security Model

The permission mechanism is built into Android and controls who can access a user's data and files, and applications following this security model must declare which permissions are required to access sensitive information, so when developing an Android app, the developer must specify in the app's manifest file which rights the app needs to function properly even though a typical user does not give much thought to the apps they install or the right they provide, and some people lack technical knowledge of Android permissions and their

ramifications (Zainab R. Alkindi et al., 2019). According to Mayrhofer et al. (2021), If an attacker can achieve all their aims within the operating system's acceptable behaviors without evading the security model, then the security mechanism is insufficient, therefore, any action that violates the aforementioned rules ought to necessitate such device-level control bypassing (rather than performing checks elsewhere, such throughout development, or even on another device). Each layer of Android relies on the one below it for security, making the Linux kernel Android's security basis, which manages processes, memory, and devices while the Hardware Abstraction Layer (HAL) works above the Linux kernel to standardize hardware interfaces; nevertheless, OS updates no longer require new hardware or reconfiguration where the HALs in a process would not have the same permissions as the rest of the process, which benefits least privileges, and while the Android's permission mechanism prevents apps from accessing private data, system resources without permission, but the apps need permission to use APIs to connect to the system; therefore, the app's manifest file should list all its rights (Mishra et al., 2022). Figure 3 demonstrates resource separation and layer accessibility

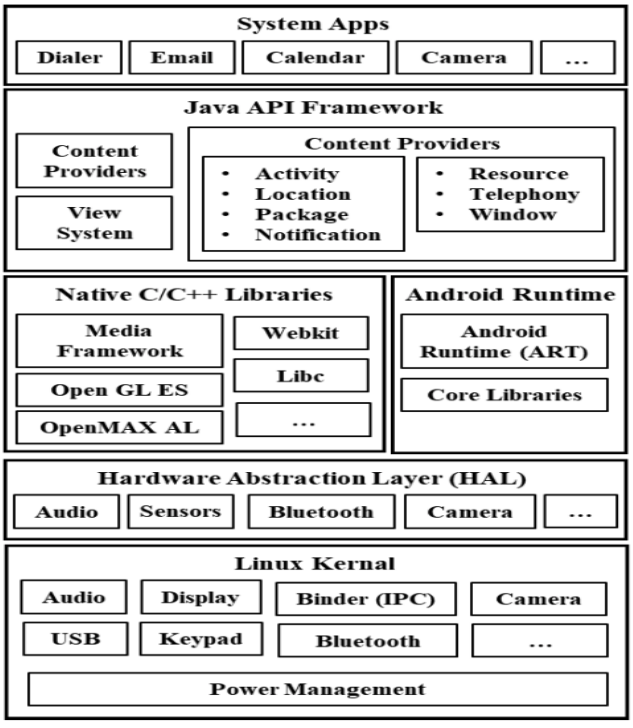
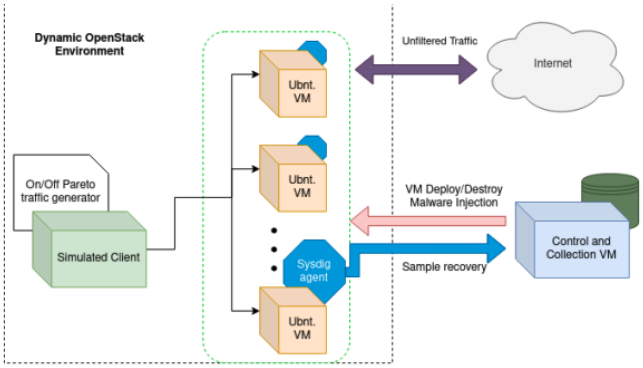


Figure 3 - Android Security Model (Mishra et al., 2022)

2.2 Android Malware Classification

Malware is any malicious software that is created with the express purpose of causing harm to a computer, server, or network. which employs malicious code to accomplish its goals; yet researchers have shown success in classifying malware by employing static analysis, dynamic analysis, and powerful deep learning algorithms (Li et al., 2022). According to Alsulami & Mancoridis (2018), the technique of determining which malware family a given malware

sample belongs to is known as malware classification where a specific malware often exhibits a set of characteristics that can be utilized to generate signatures for detection and categorization and extracting a signature might classify it as either static or dynamic. Malware under investigation both during and after execution is known as "dynamic analysis" while in contrast to static analysis, dynamic analysis involves actually running the malware and observing its results. (Brown et al., 2022). Furthermore, Kumar et al. (2018) also explained that malware analysis primarily uses static analysis without running the malware, which makes it safe; however, dynamic analysis observes the malware sample running it in a controlled or isolated environment, and it can be analyzed using several methods such as Reverse engineering, Debugging, Disassembler, Packers (runtime).



ALL CLASSES FROM THE 4,180 FINAL SAMPLES

Class	Count	Used
trojan	2299	✓
virus	616	✓
backdoor	382	✓
rootkit	253	✓
miner	226	✓
grayware	142	✓
worm	142	✓
none	87	
ransomware	21	
downloader	7	
bot	3	
hoax	2	

Figure 4 - Malware Analysis performed on an OpenStack environment (Brown et al., 2022).

2.3 Android Malware Analysis Methods and Tools

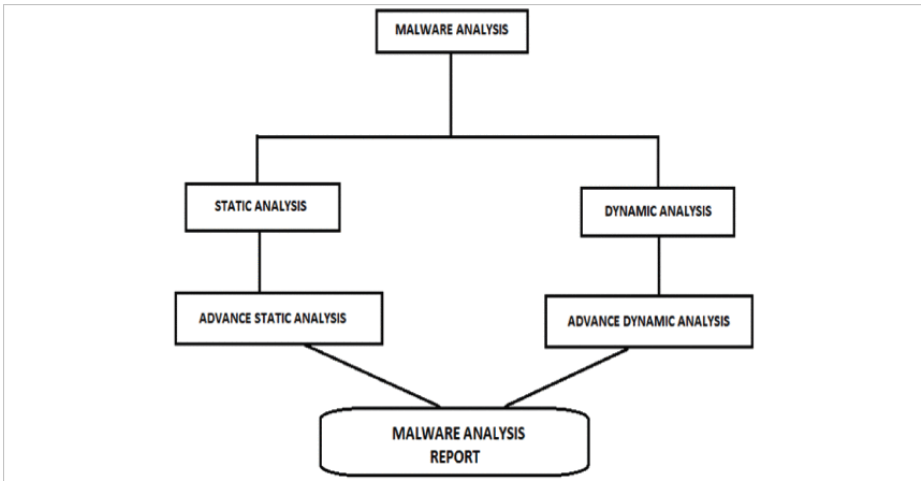


Figure 5: Malware Analysis Methods

The Android OS is a widely used and very well-liked operating system. As stated by Hadiprakoso et al. (2020), Android allows users to download apps from unofficial stores and install them, which gives cybercriminals the opportunity to infiltrate Android smartphones with malware. Even though static and dynamic malware analysis and detection technologies have

been established, current research still falls short of optimal performance, while Tahtaci & Canbay (2020) stated that malware can compromise a system to the point where sensitive information like login credentials and credit card numbers are stolen, and human effort may not be sufficient in the fast-expanding malware market despite the existence of antivirus software and malware analysis teams. In relation to Arif et al. (2021), static analysis involves five steps: data collection, reverse engineering tools, feature extraction, feature selection, and feature evaluation. Figure 6 depicts a static analysis of Android malware, which first collects malicious and harmless datasets, and recent studies show that using more datasets improves experiment accuracy, but researchers evaluate their proposed approaches using Drebin, Genome, VirusShare, Contagio, AMD, Kaggle, Androzoo, and Anzhi benign datasets; they often use Google Play; however, they reverse-engineer the apk file using PKtool, Androguard, FlowDroid, Aapt, Baksmali, Soot, Dex2jar, and Dedexer.

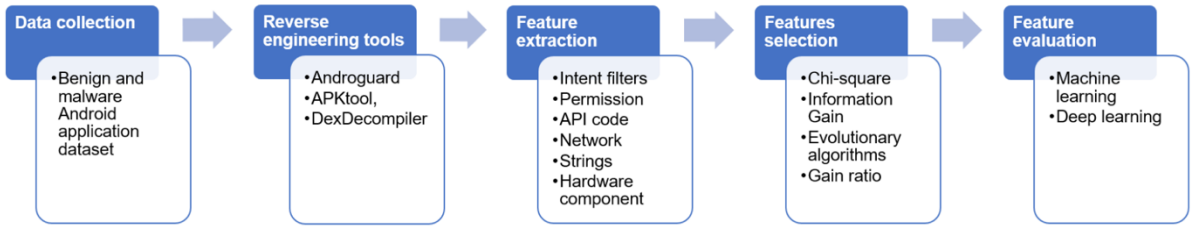


Figure 6 - Method for Static Analysis of Android Malware (Arif et al., 2021)

Figure 7 below shows the comparison between static and dynamic analysis.

	Static analysis	*Dynamic analysis*
Procedure	Analyse code without executing it.	Analyse the behaviour of the application while it is executing.
Tools	Reverse engineering tools+analysis tools/manual analysis.	Tools to run the app+analysis tools.
Accuracy	More accurate	Less accurate.
Disadvantage	• Unavailability of source code makes it harder. • Does not analyse unknown malware. • Need to know malware patterns in advance. • Time consuming. • Obfuscation issues. • High false positive rates (Mendonça et al., 2013).	• Slow. • Unsafe. • Resource consuming. • Incomplete code coverage. • Cause an inexact behaviour log of the malware.
Advantages	• Fast and safe. • Not very resource consuming. • Complete analysis of a program. • Covers all possible execution paths.	• Detects unknown malware. • Avoid obfuscation issues.

Figure 7: Static and Dynamic Analysis Comparison (Rani & Dhindsa, 2016)

As explained by Vurdelja et al. (2020), it is possible for the execution environment to be damaged when performing dynamic malware analysis, this relies on running malicious code, but a robust sandboxing architecture is necessary to ensure accurate analysis and shield the computer system and smartphone devices from malware's destructive effects while in advance

dynamic analysis, further analysis is done including the basic dynamic analysis and Qbeitah & Aldwairi (2018), stated that in dynamic method, the programme is actively used to study its detection strategies, habits, and features. Since all programmes must be executed, this method consumes a lot of CPU time. Tools like VirtualBox, Sandboxes, Resource Hacker, Process Hacker, Wireshark, and Immunity Debugger are used for dynamic analysis; while PEid, PEView, MD5Deep, VirusTotal.com, and Ida pro, and Dependency Walker are used for static analysis; the analysis report is generated in either case (Kumar et al., 2018). Figure 8 shows a flowchart of dynamic analysis.

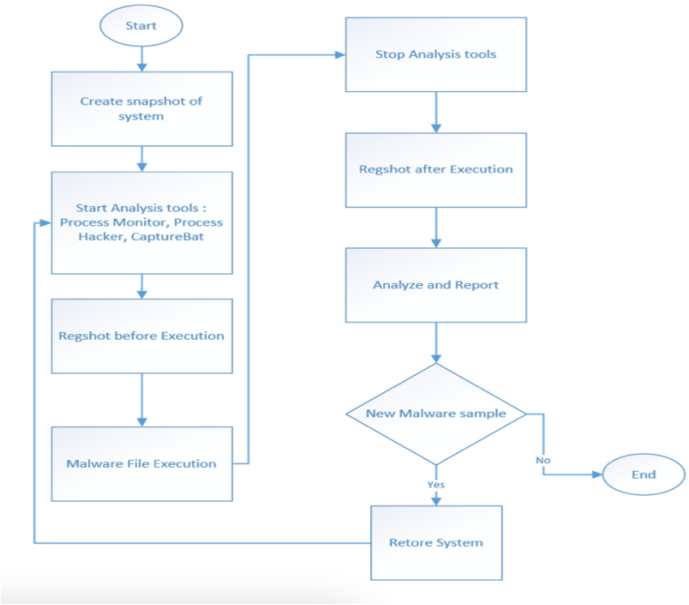


*Figure 8 - Flowchat for Dynamic Malware Analysis (*Qbeitah & Aldwairi, 2018)

2.4 Android Malware Infection Method

Malicious software, or "malware," has become more of a problem as the number of people using the Internet has risen, making malware easier to compromise our privacy and lead to financial loss or other negative consequences (Rani & Dhindsa, 2016b). Agreeing to Microsoft (2022), malware can infect the devices by exploiting a known software vulnerability, which is similar to a slot in the software that allows malware unauthorized access to the device. However, a website you visit could try to exploit a vulnerability in your web browser to infect the computer and smartphone with malware. Also, Panda Security (2022) stated that social networks give cybercriminals a platform to spread their work; however, on Facebook, Twitter, and WhatsApp, email was the primary tool for sending spam, but these platforms allow fake accounts to be used to send chain letters and malicious links, share inappropriate content, and spoof user identities, while phishing scams are used to trick users into clicking on infected links.

2.4.1 Android Analysis Environment

As demonstrated by Faghihi et al. (2022), techniques to prevent reverse engineering which includes things like string encryption, control-flow obfuscation, dead-code injection, and more are an effective technique for detecting malware in signature matching and other forms of static analysis and not in dynamic analysis because signature matching explores only for areas where well known malicious codes are, whereas dynamic analysis runs an applications that are malicious in an isolation environment(Sandbox) and analyzes their actions, which makes dynamic analysis useful for countering the aforementioned methods, but malware can detect sandboxes and avoid their malicious actions while being analyzed. Furthermore, agreeing with Sihag et al., (2021) who explained that an Android application's memory and resources are compartmentalized in a kernel-level sandbox, and due to the sandbox's location in the kernel, applications (OS libraries, applications, and frameworks) are isolated from one another and cannot access one another's resources or perform any malicious actions which safeguard both developer and system apps against malicious software, and while applications that are signed with the developer's private key share the identical unique ID (sandbox) for resources and permissions, the malware creator with a developer's key can develop an application with an identical certificate to access sibling applications' private resources (for example, Contacts) if the application has access to those resources.

2.4.2 Android Malware Detection and Prevention Methods

Signature-based and Permission-based approaches are two common types of malware detection methods. However, Signature-based malware detection is used by commercial antimalware programmes and uses semantic patterns to generate a fingerprint that makes the malicious software identifiable by comparing its signature to known malware families, Code obfuscation is a common method used to avoid signature-based detection systems, which can be fooled by previously unknown malware variants that could be prevented by urgently updating the malware to uncover new malware variants resulting from repackaging and code obfuscation (Gillani, 2022). In contrast, Gillani (2022) explained that AndroSimilar employs a statistical signature technique in which access controls in Android are managed by app permissions using permission-based detection analysis where users must provide permission to access all app features during installation and AndroidManifest.xml is where developers declare which resources the application is allowed to access, making it difficult to detect malicious behaviour because developers frequently declare which resources the application is allowed to access. Once malware has been discovered, the National Cyber Security Centre (2022) recommends the following five (5) steps be taken to prevent further infection of malware as shown in Table One (1) below:

Table 1 - Malware prevention steps (National Cyber Security Centre, 2022)

1.	Virus protection software, like anti-virus software, should be used to stop viruses from spreading.
2.	staying away from downloading any suspicious applications.
3.	Keeping all of the devices' software and firmware at the most recent, stable versions (patching and updating).
4.	The use of memory cards and USB drives should be restricted.
5.	Turn on your firewall at all times.

2.4.3 Tools for Generating Malware String Signature for Future Detection

Software designed to detect and eliminate malware, such as viruses, malicious browser helper objects, browser hijackers, spyware, etc., is essentially a fast-string search program, and these programmes protect the systems from social engineering and other potential threats; however, virus signatures are commonly used by antivirus software during the scanning process, which is typically carried out in a sequential fashion (Sahoo et al., 2015). As reported by (Feng et al., 2017), Syntactic, semantic, or both types of signatures are possible for generating string signatures for future malware detection. APPOSCOPY identifies Android malware using a semantic signature, which uses static taint analysis and intercomponent control flow analysis to match signatures to Android applications; however, ASTROID is a plug-in for APPOSCOPY that generates malware signatures, and while KIRIN uses Android permissions to detect malware that classifies applications as benign or malicious using permission patterns, FACT uses dynamic analysis to find a signature, while ASTROID uses static analysis to find a maximally weighted common subgraph (Feng et al., 2017).

2.5 Conclusion

This session delves into the Android architecture and how it all comes together to meet the needs of Android devices, as well as how Android devices are susceptible to malware attacks since users download unsigned applications from numerous sources without thinking about the security of the applications they are installing. It also discussed how dynamic malware analysis techniques could find every DLL that malware imports, the process malware uses to run inside the system, how it operates, and every network traffic connection it makes to perform its malicious activities. In contrast, the static analysis method can reveal more comprehensive details about the malware's structure, features, and aptitudes, such as its ability to corrupt other programmes, make changes to the registry, and create new files and directories. In addition, this research details the various static and dynamic malware analysis techniques, tools, and

environments that can be used to uncover this malware and detect and prevent malicious activities. Finally, to safeguard end users' devices from malicious software and attacks, it is suggested that static, dynamic, and behavioural malware analysis be combined and embedded into the Android operating system, along with an application that guides users through improving malware detection and prevention.

References

Alatawi, H., Alenazi, K., Alshehri, S., Alshamakhi, S., Mustafa, M. & Aljaedi, A. (2020). Mobile Forensics: A Review. In: *2020 International Conference on Computing and Information Technology, ICCIT 2020*. 9 September 2020, Institute of Electrical and Electronics Engineers Inc.

Alsulami, B. & Mancoridis, S. (2018). *Behavioral Malware Classification using Convolutional Recurrent Neural Networks*. [Online]. Available from: http://arxiv.org/abs/1811.07842.

Alzaabi, M. (2013). Ontology-based forensic analysis of mobile devices. In: *Proceedings of the IEEE International Conference on Electronics, Circuits, and Systems*. 2013, Institute of Electrical and Electronics Engineers Inc., pp. 64–65.

Anna University (2022). *CS8493-UNIT-5-Android-Architecture.pdf*. [Online]. 2022. Anna University. Available from: https://www.binils.com/wp-content/uploads/2021/11/CS8493-UNIT-5-Android-Architecture.pdf. [Accessed: 9 November 2022].

Arif, J.M., Razak, M.F.A., Awang, S., Tuan Mat, S.R., Ismail, N.S.N. & Firdaus, A. (2021). A Review: Static Analysis of Android Malware and Detection Technique. In: *Proceedings - 2021 International Conference on Software Engineering and Computer Systems and 4th International Conference on Computational Science and Information Management, ICSECS-ICOCSIM 2021*. 1 August 2021, Institute of Electrical and Electronics Engineers Inc., pp. 580–585.

Brown, P., Brown, A., Gupta, M. & Abdelsalam, M. (2022). *Online Malware Classification with System-Wide System Calls in Cloud IaaS*. In: 8 September 2022, Institute of Electrical and Electronics Engineers (IEEE), pp. 146–151.

Chen, Y. (2021). Research on Android Architecture and Application Development. In: *Journal of Physics: Conference Series*. 25 August 2021, IOP Publishing Ltd.

Da Costa, A.M., de Sa, A.O. & Machado, R.C.S. (2022). Data Acquisition and extraction on mobile devices-A Review. In: *2022 IEEE International Workshop on Metrology for Industry 4.0 and IoT, MetroInd 4.0 and IoT 2022 - Proceedings*. 2022, Institute of Electrical and Electronics Engineers Inc., pp. 294–299.

Dorai, G., Houshmand, S. & Aggarwal, S. (2020). Data extraction and forensic analysis for smartphone paired wearables and IoT devices. In: *Proceedings of the Annual Hawaii International Conference on System Sciences*. 2020, IEEE Computer Society, pp. 1401–1410.

Dovydas Patapas (2021). *Investigation of Digital Forensic Methods for Mobile Devices*.

Faghihi, F., Zulkernine, M. & Ding, S. (2022). CamoDroid: An Android application analysis environment resilient against sandbox evasion. *Journal of Systems Architecture*. 125.

Feng, Y., Bastani, O., Martins, R., Dillig, I. & Anand, S. (2017). *Automated Synthesis of Semantic Malware Signatures using Maximum Satisfiability*. In: 13 May 2017, Internet Society.

Institute of Electrical and Electronics Engineers (2022). *SoutheastCon 2018 : St. Petersburg, FL., Apr 19th - Apr 22nd, 2018.*

Kumar, S., Lakshmi Narain College of Technology, Institute of Electrical and Electronics Engineers & Institute of Electrical and Electronics Engineers. Bombay Section. Madhya Pradesh Subsection. (2018). *2018 International Conference on Advanced Computation & Telecommunication : ICACAT - 2018 : 28-29 December 2018.*

Li, L., Ding, Y., Li, B., Qiao, M. & Ye, B. (2022). Malware classification based on double byte feature encoding. *Alexandria Engineering Journal*. 61 (1). p.pp. 91–99.

Li, X., Ortiz, P.J., Browne, J., Franklin, D., Oliver, J.Y., Geyerz, R., Zhou, Y. & Chong, F.T. (2010). Smartphone evolution and reuse: Establishing a more sustainable model. In: *Proceedings of the International Conference on Parallel Processing Workshops*. 2010, pp. 476–484.

Mayrhofer, R., Stoep, J. vander, Brubaker, C. & Kralevich, N. (2021). The Android Platform Security Model. *ACM Transactions on Privacy and Security*. 24 (3).

Microsoft (2022). *How malware can infect your PC - Microsoft Support*. [Online]. 2022. Microsoft. Available from: https://support.microsoft.com/en-us/windows/how-malware-can-infect-your-pc-872bf025-623d-735d-1033-ea4d456fb76b. [Accessed: 11 November 2022].

Mishra, B., Agarwal, A., Goel, A., Ansari, A.A., Gaur, P., Singh, D. & Lee, H.N. (2022). Privacy Protection Framework for Android. *IEEE Access*. 10. p.pp. 7973–7988.

Moss, S. (2021). From "brick" to smartphone: the evolution of the mobile phone. *MRS Bulletin*. 46 (3). p.pp. 287–288.

Panda Security (2022). *Main infection techniques - Panda Security*. [Online]. 2022. Panda Security. Available from: https://www.pandasecurity.com/en/security-info/infection-techniques/. [Accessed: 11 November 2022].

Phongtraychack, A. & Dolgaya, D. (2018). Evolution of Mobile Applications. In: *MATEC Web of Conferences*. 28 February 2018, EDP Sciences.

Qbeitah, M.A. & Aldwairi, M. (2018). Dynamic malware analysis of phishing emails. In: *2018 9th International Conference on Information and Communication Systems, ICICS 2018*. 4 May 2018, Institute of Electrical and Electronics Engineers Inc., pp. 18–24.

Rani, S. & Dhindsa, K.S. (2016a). Malware detection techniques and tools for Android. *International Journal of Social Computing and Cyber-Physical Systems*. 1 (4). p.p. 326.

Rani, S. & Dhindsa, K.S. (2016b). Malware detection techniques and tools for Android. *International Journal of Social Computing and Cyber-Physical Systems*. 1 (4). p.p. 326.

Rao, V.V. & Chakravarthy, A.S.N. (2016a). Forensic analysis of android mobile devices. In: *2016 International Conference on Recent Advances and Innovations in Engineering, ICRAIE 2016*. 2016, Institute of Electrical and Electronics Engineers Inc.

Rao, V.V. & Chakravarthy, A.S.N. (2016b). Forensic analysis of android mobile devices. In: *2016 International Conference on Recent Advances and Innovations in Engineering, ICRAIE 2016*. 2016, Institute of Electrical and Electronics Engineers Inc.

Sahoo, A.K., Sahoo, K.S. & Tiwary, M. (2015). Signature based malware detection for unstructured data in Hadoop. In: *2014 International Conference on Advances in Electronics, Computers and Communications, ICAECC 2014*. 6 January 2015, Institute of Electrical and Electronics Engineers Inc.

Schobel, J., Volz, M., Hörner, K., Kuhn, P., Jobst, F., Schwab, J.D., Ikonomi, N., Werle, S.D., Fürstberger, A., Hoenig, K. & Kestler, H.A. (2021). Supporting medical staff from psycho-oncology with smart mobile devices: Insights into the development process and first results. *International Journal of Environmental Research and Public Health*. 18 (10).

Sihag, V., Vardhan, M. & Singh, P. (2021). A survey of android application and malware hardening. *Computer Science Review*. 39.

StatCounter (2023). *Mobile Operating System Market Share Worldwide | Statcounter Global Stats*. [Online]. 2023. StatCounter © 1999-2023. Available from: https://gs.statcounter.com/os-market-share/mobile/worldwide. [Accessed: 14 January 2023].

Tajuddin, T.B. & Manaf, A.A. (2015). Forensic investigation and analysis on digital evidence discovery through physical acquisition on smartphone. In: *2015 World Congress on Internet Security, WorldCIS 2015*. 16 December 2015, Institute of Electrical and Electronics Engineers Inc., pp. 132–138.

Toppo, P. & Dhote, T. (2021). PREFERENCE OF MOBILE PLATFORMS: A STUDY OF iOS VS ANDROID. *International Journal of Modern Agriculture*. (10).

Vurdelja, I., Blazic, I., Bojic, D. & Draskovic, D. (2020). A framework for automated dynamic malware analysis for Linux. In: *2020 28th Telecommunications Forum, TELFOR 2020 - Proceedings*. 24 November 2020, Institute of Electrical and Electronics Engineers Inc.

Wang, H., Li, R., Shokouhi, M., Li, H. & Chang, Y. (2017). Search, mining, and their applications on mobile devices: Introduction to the special issue. *ACM Transactions on Information Systems*. 35 (4).

Woo et al. (2022). *Patent Application Publication*.

Zainab R. Alkindi, Mohamed Sarrab & Nasser Alzidi (2019). Proceedings. *Free and Open Source Software Conference*.

Zinkus, M., Jois, T.M. & Green, M. (2021). *Data Security on Mobile Devices: Current State of the Art, Open Problems, and Proposed Solutions Executive Summary*.