Journal of
Cybersecurity and Privacy

MDPI

Article

Exploratory Study on Kali NetHunter Lite: A Digital Forensics Approach

Miloš Stanković and Umit Karabiyik *

Department of Computer and Information Technology, Purdue University, West Lafayette, IN 47907, USA
* Correspondence: umit@purdue.edu; Tel.: +1-765-496-6877

Abstract: Mobile devices, specifically smartphones, have become a necessity in everyday life, as we perform many essential day-to-day tasks using these devices. With the projected increase in mobile devices to 18.22 billion by 2025, the reliance on smartphones will only grow. This demand for smartphones has allowed various companies to start developing their own devices and custom operating systems, each of which puts its own touch on them. In addition, current smartphones have increased processing power, providing users with a computer experience in their pockets. Software developers have taken this opportunity to bridge the gap between personal computers and smartphones by creating the same software for personal computers and mobile devices. Kali Linux is one of the most popular penetration testing tools for desktop use and has been adapted to operate on mobile devices under the name *Kali NetHunter*. Kali NetHunter has three different versions on mobile platforms that provide various levels of capabilities. Kali NetHunter is just one example in which an application or an operating system applies to a specific niche of users. Highly customized operating systems or applications do not receive the same attention as field research, leaving them unfamiliar to mobile forensic investigators when used maliciously. In this paper, we conducted an exploratory study on the Kali NetHunter Lite application after it was installed and its embedded tools were utilized. Our results show a detailed analysis of the file system and reveal the data from the tests carried out during various phases. Furthermore, the locations of the folders involved in the process were described.

Keywords: Kali NetHunter Lite; Android; iOS; mobile forensics; digital forensics

Citation: Stanković, M.; Karabiyik, U. Exploratory Study on Kali NetHunter Lite: A Digital Forensics Approach. *J. Cybersecur. Priv.* **2022**, *2*, 750–763. https://doi.org/10.3390/jcp2030038

Academic Editors: Mario Antunes and Carlos Rabadão

Received: 11 August 2022
Accepted: 15 September 2022
Published: 19 September 2022

1. Introduction

The dependency on mobile devices, particularly smartphones, has increased dramatically over the last decade. According to [1], the number of mobile devices will increase from 14.91 billion in 2021 to 18.22 billion in 2025. With so many units on the market, the diversity of devices and operating systems (OSs) allows users to choose what is most appropriate for their needs. Of all the different OSs, Android is the most common, with approximately 70% of mobile devices, followed by iOS with around 28% [2]. Android OS and its variations are installed on approximately 73% of mobile devices worldwide [3]. Over the years, Android has evolved and branched into many different iterations, becoming the operating system for various smartphone and mobile phone manufacturers. Additionally, technology advancements have allowed smartphones to have the processing power to the extent of a personal computer or a laptop, ultimately aiding the user but also allowing malicious users to gain access to the user's data through another source. With that much processing power, hackers using mobile devices can exploit users by mimicking commonly used applications and gaining access to location, photos, and other information. When devices used in those crimes are captured, they create a problem for mobile forensic investigators due to their modified OS.

On top of the modified OSs and frequent updates, custom applications add another layer of difficulty to the process. For example, one of the most widely used penetration testing platforms, Kali Linux [4], is available on mobile devices under the name Kali NetHunter.

Kali NetHunter [5] is an open-source Android platform created for penetration testers who need a portable version of Kali Linux without sacrificing many features. Over the years, Kali NetHunter has developed three different versions, NetHunter Rootless, NetHunter Lite, and NetHunter [5]. The Lite version has almost all the capabilities as the full version (NetHunter), except for WiFi Injections and HID Attacks [5]. Only requiring the device to be rooted (i.e., providing privileged user access), the Lite version of NetHunter provides the most flexibility without looking for a specific device that can support the full version. Despite the actual intention of the tool and its capabilities, Kali NetHunter can also be used for malicious purposes. This is a concern for digital forensics, particularly mobile forensic investigators, as there is little analysis performed on the application.

As seen in recent studies [6–10], it is not unusual to have a comprehensive investigative study on mobile operating systems or applications. Regardless of the capabilities and potential of the Kali NetHunter platform, there is a lack of publicly available research on the topic. The review of related work shows that the platform is being used to carry out various attacks [11–13], but no analysis has been performed on the application itself.

This research aims to close the literature gap and perform an exploratory study of Kali NetHunter Lite and its tools (e.g., MAC Changer, USB Arsenal, HID Attacks) that are installed on the OnePlus 7T smartphone using digital forensic analysis techniques. By following digital forensics methods (obtaining forensics data, preparation, identification, analysis, and reporting), our goal is to address the problem of encountering targeted operating systems and smartphones during the course of digital forensics investigations. Therefore, this study is divided into two phases to represent all aspects of possible real-life scenarios where Kali NetHunter Lite might be used, such as criminal activity or penetration testing. The first phase involves rooting the device and installing Kali NetHunter Lite. During the second phase, the tools provided by NetHunter Lite are utilized; later, the physical image of the smartphone is analyzed. Therefore, the contributions of this paper include the following:

- Providing a detailed analysis of the NetHunter Lite application right after it has been installed.
- Providing a detailed analysis of the NetHunter Lite application after the tools provided have been used.
- Highlighting the main locations of the NetHunter Lite application in the file structure for the use of digital forensics investigators.

The remainder of this paper is structured as follows. Section 2 outlines the work related to this study. Section 3 provides and explains the methodology used to perform the experiment. Section 4 presents the findings of the two phases, and Section 5 discusses the findings and the challenges faced in this research. Lastly, Section 6 provides a conclusion of the experiment and discusses future work based on the results.

2. Related Work

Kali NetHunter is an open-source mobile operating system for Android devices designed to help penetration testers during security evaluations. Kali NetHunter comes in three different variations, NetHunter Rootless, NetHunter Lite, and NetHunter [5]. Current research lacks forensic examination and analysis of the operating system and its variations. In the absence of related literature, this paper looked at the literature on similar niche mobile operating systems. The research also incorporated an official Kali NetHunter website [5], where most of the instructions and documentation for the use of the installation are located.

Custom-designed operating systems and mobile devices are not new to the market. As explored in [10], the Ubuntu Touch OS (Operating System) installed on PinePhone created by Pine Microsystems is a perfect example of today's diversity of software and hardware. In the paper [10], researchers wanted to expand the understanding of Ubuntu Touch OS, list important artifacts and their locations, and test forensic tools combined with the phone and OS. By completing the file structure and analyzing apps, the researchers

wanted to help mobile forensic investigators in future cases when needed. Due to the analysis being different from any Android or Linux devices, the result of the experiment was a detailed course for the investigators to take if ever faced with a similar issue. The process consisted of three main phases, the installation of Ubuntu Touch OS, the data population, and the analysis [10]. During data population, various applications were installed and populated. The applications resembled everyday use (e.g., messaging, browsing the web, and sending photos). In the analysis phase, researchers found that the Ubuntu Touch OS is almost identically organized as the Ubuntu OS for desktop computers [10]. The researchers concluded that the applications installed on the device did not reveal much information as a result of the applications that use online storage and the Ubuntu Touch OS being still in the development phase [10].

Tzvetanov and Karabiyikn demonstrated a road map to a new mobile operating system, SailfishOS. In this study, the researchers explored mapping the artifacts found in the SailfishOS 3.2 filesystem. Furthermore, the researchers presented different OS acquisition techniques. A significant obstacle to the study was the lack of literature on the operating system and its functioning during the research. To overcome the problems, the authors turned to mobile forums and an older publication that showed the acquisition of the previous version of SailfishOS. Despite the challenges, the study was able to show a significant contribution to the target audience. The methodology used to complete the study involved the setup of the hardware, installation of the OS, acquisition, and analysis [8]. The acquisition steps were divided into two steps, in which the evidence acquisition was performed using open-source and commercial software [8]. Lastly, the analysis was divided into phone changes that occurred during the process and while examining the process. The authors discovered how to acquire data from SailfishOS 3.2 utilizing both commercial and open-source tools. Furthermore, the article offered locations for significant artifacts used during digital forensic investigation [8].

One research paper [14] analyzed malware applications on Android devices utilizing two different types of tools, one static and one dynamic. The research analyzed source code in the search for any malicious software and found that Flowdroid performed the best. Many researchers have attempted to analyze phones for malicious software but limited work was found where the whole mobile device was a potential malicious adversary. The reason for this is because no operating system on the mobile phone is capable of containing applications that are used for penetration testing. Examples of those tools are shown in another research [11], where multiple wireless attacks were carried out to crack passwords, and Kali NetHunter was used. The researchers used applications such as aircrack-ng, airmon-ng, and airodump-ng to attack a router through the network. The process was performed to expose the WPA and WPA-w vulnerabilities.

In [15–19], various attacks such as DoS (Denial of Service), DDoS (Distributed Denial of Service), MITM (Man in the Middle), and many more were performed on the selected smartphones. For example, in [15], the researchers showed different phones with two different operating systems (iOS and Android) being attacked via DoS and DDoS methods. Similarly to the previous study, [16] shows the vulnerabilities of the smartphones being exploited throughout the Wi-Fi connectivity protocols. Both studies show the vulnerabilities of two different platforms of smartphones. In [17], a whole botnet network of mobile devices was created to execute the DoS attack while proposing a solution for mitigating the problem. Approaching the problem from a different perspective, study [18] implemented a DoS attack in a test case called 'SlowDroid', implementing elements to render the network utilization useless. The researchers concluded that even one mobile device capable of carrying out a DoS attack could tremendously slow down the network. On the contrary, the research [19] attempted to create an IDS (Intrusion Detection System) where the attack on the smartphone can detect a potential attack by deviating from the CPU's (Central Processing Unit) patterns. While these studies have shown how smartphones can be used in various attacks and their services being disrupted, the literature lacks forensics studies in situations where the smartphone is the device attacking other smartphones.

Carthy et al. showed the importance of understanding technologies and practices when it comes to digital investigation. While the paper goes into depth on approaches when it comes to the digital forensics process and the steps, the paper shows a case study in which Kali NetHunter was configured for auditing of a remote server. Kali NetHunter was installed on a Nexus 6 device with ROM (Read-Only Memory) from 2017 [12].

Having good tools when performing security assessments is essential. The study on identifying attacks on smart TVs in addition to Samsung was no exception to this [20]. In this study, the researchers used an Asus Nexus 7 with an Android operating system with Kali NetHunter 1.2 installed for penetration testing. The researchers found that users of smart TVs are easy targets for an attacker [20]. The attack was performed using the HID Ducky script attack provided by Kali NetHunter. This type of attack creates a payload that can be injected into a targeted device pretending to be a keyboard that enters the keys.

Utilizing Kali NetHunter for malicious uses is also possible, as seen in [13]. For this experiment, an attacker utilized a feature of NetHunter called Metasploit Framework used for security and vulnerability assessments. During the assessment, the attacker was able to track the phone and take over the camera and other sensitive information [13]. As seen in this experiment, Kali NetHunter can be utilized for malicious activities, which makes the operating system very important for digital investigators.

As seen in the literature, there have been some uses of Kali NetHunter over the years, but no work has been conducted on the examination of the system itself. Most of the uses included Kali NetHunter as a penetration testing tool, but the literature did not show that the system was forensically analyzed for its structure.

3. Methodology

As the number of mobile devices continues to grow, so will the number of applications and operating systems that use them. Although diversity is suitable for users, it creates challenges for digital forensic investigators. Considering that many people today carry a mobile device, leile devices play an important role in both criminal and civil investigations [21]. Diversified operating systems and applications on mobile devices change rapidly, with each update expanding the complexity of the already challenging situation for forensic investigators [22]. Kali NetHunter is an application overlay that has the ability to run on multiple versions of Android and on multiple phone models. Belonging to the family of Linux Kali OS, Kali NetHunter is capable of performing actions related to penetration testing, such as evil access points, man-in-the-middle attacks, and many more. This article focused on exploring Kali NetHunter Lite, ultimately aiding digital forensic investigators in their challenges and enriching the currently available knowledge.

During the research, various methods and software were used to complete the analysis. Some applications such as ADB (Android Debug Bridge), MsmDownloadTool, Magisk, and others were necessary to deploy Kali NetHunter Lite on the mobile phone. The remaining software was used for the creation of the physical images and the analysis process. The research focused on commercial tools only for the examination of the artifacts. The full list of software used is listed in Table 1. Lastly, the experimental methodology was divided into multiple steps. The steps presented below ensured the consistency and progress of the project.

3.1. Environment Preparation

The first step of the process was to ensure that all devices used were reset and the previous data were removed. Deployment of the new software (Windows 10 and Kali Linux) on the laptops from Table 2 was accomplished as well as resetting the smartphone to default settings. The operating system on the examination PC was not reinstalled since it contained the examination and analysis software; further, not installing a new version of the OS does not affect the study by any means. Lastly, the wireless router (TP-Link Wireless Router) was also reset to factory settings.

Table 1. List of the software used for the experiment.

Software Name	Version	Usage
Android	10	StockOS (10.0.13.HD63CB)
Oxygen OS	10	InstalledOS (10.0.6.HD65AA)
ADB	1.43	Rooting
MsmDownloadTool	4.0	Rooting
Magisk	23.0	Rooting
Root Checker Basic	6.5.0	Rooting Check
SuperSU	1.0.1	Rooting Check
NetHunter Store	2019.3	Application
NetHunter	2021.3	Application
NetHunter Terminal	2020.4	Application
Magnet AXIOM Process	5.6.0.26839	Acquisition
Magnet AXIOM Examine	5.6.0.26839	Examination and Analysis
Cellebrite UFED 4PC	7.42.0.82	Acquisition
Cellebrite Physical Analyzer	7.42.0.50	Examination and Analysis

Table 2. Equipment used for the experiment.

Devices (Name and Model)	Version	Device ID
HP Laptop (Kali Linux)	2020.4	Pavilion DV6
Asus Laptop (Windows 10)	20H2 (19042.1110)	G570
OnePlus 7T	HD1907	418fb866
TP-Link Wireless Router	2.1	TL-WR940N
Examination PC (Windows 10)	21H1 (19043.1348)	Optiplex 640

The second step included configuring the wireless router, ensuring connectivity to the Internet for any dependencies. All local devices were assigned a unique IP address, and those were not changed throughout the study (Figure 1). Since some of the devices had only wireless capabilities, the router's SSID (Service Set Identifier) was set to NetHunter with a password requirement and WPA-2 security.

Upon successful configuration of the network, an existing email (p*****@gmail.com) was designated for the project. The account was used for any profile and/or account registration. Figure 1 depicts the entire network and device setup used in this experiment.

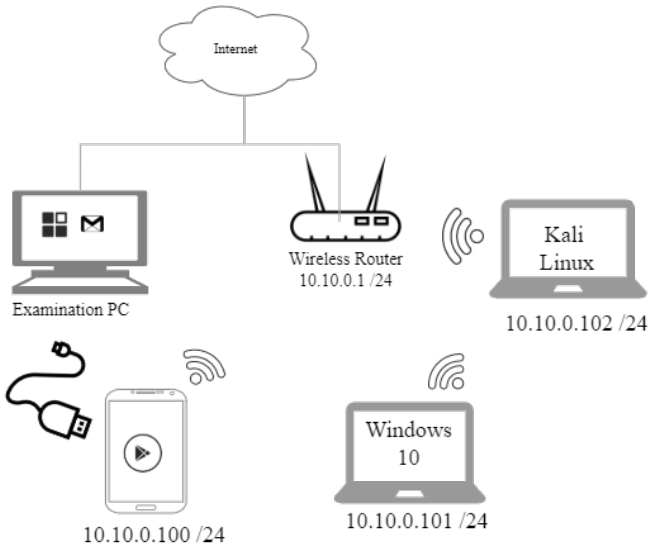


Figure 1. Devices and network setup.

3.2. Mobile Phone Preparation

Although the OnePlus 7T smartphone was unlocked when purchased, the bootloader was not. Obtaining the bootloader file required creating an account on the OnePlus website confirming the IMEI (International Mobile Equipment Identity) number and submitting the request for the file. The process was unique to T-Mobile phones and took approximately seven days. Once the file was downloaded, the ADB tool and drivers were installed and the file was copied to the ADB root folder. After activating *developer mode* on the OnePlus 7T, the phone was booted into the bootloader. The phone was connected to the laptop via a cable, and the ADB command prompt was opened. To unlock the phone and transfer the unlock file, the following commands were executed:

- *fastboot flash cust-unlock unlock.bin;*
- *fastboot oem unlock.*

The last step in unlocking the bootloader was to confirm the *unlock* and *restart* the phone.

In order to install Kali NetHunter Lite, it was recommended to have full root access to the phone. Obtaining root access to the phone ensures that the user gains privileged access to the OS. Since the test phone had a custom operating system proprietary to T-Mobile (Android 10), it was necessary to convert it to OxygenOS. OxygenOS is a less restricted version of the OS that can be installed on the phone and for which it is easier to gain root access. Swapping to a different OS required downloading the correct version of OxygenOS and utilizing the MsmDownloadTool V4.0 tool. Once the software installed the new OS, the next step was to install Magisk and root the phone. The phone was connected to the Wi-Fi network (NetHunter) and obtained an internet connection before downloading the latest available version of Magisk. Upon downloading and installing Magisk, the phone was restarted and two more applications were downloaded. SuperSU and Root Checker Basic were the applications installed to ensure that the rooting of the phone was performed successfully. Both applications confirmed that the phone was properly rooted.

The last step of the phone preparation was to install NetHunter. To install it, first, an application called NetHunter Store needed to be downloaded. Upon installing and opening the app, the NetHunter application was presented and downloaded. The NetHunter application needed two dependencies (Kali Chroot Manager and NetHunter Terminal) to be installed before it could run correctly. The installation of NetHunter Terminal was performed throughout the NetHunter Store app and the installation of Kali Chroot Manager was initialized in the NetHunter application itself. Once the download was completed and Kali Chroot Manager was running, the phone was ready to utilize Kali NetHunter's capabilities.

3.3. Data Population

The data population phase consisted of running the tools provided by the NetHunter application. The procedure included running all processes from top to bottom. Some of the attacks required additional devices on and off the network (e.g., DuckHunter HID, APACHE2, SSH). Table 3 shows all the services provided by the NetHunter application and whether researchers were able to perform them successfully. As noted in the table, *YES** means that the services were attempted but the completion was not obtained. For instance, the service 'DNSMASQ' ran, but at the time of the project, there was no way to check if the process was successful or not. Moreover, the 'DeAuth' service did not provide any results, but the researchers created a whitelist with an IP Address in it. It is also worth noting that the databases in Kali Services, Custom Commands, and USB Arsenal were backed up. The attempts marked *NO* in Table 3 were either not successful or could not be performed due to the limitations of the phone and the timeline of the project. For instance, POSTGRESQL showed an error message 'Failed Starting POSTGRESQL service'; Mana Wireless Attack failed to generate the certificates; Bluetooth arsenal interface did not recognize the phone's Bluetooth, giving the error 'no interfaces selected'; SearchSploit never passed the screen 'feeding exploit DB'. The 'Kex Manager' was not installed and

the researchers decided not to pursue it. Additionally, when 'MITM Framework' was attempted, the terminal showed no commands found. Challenges during the experiment will be discussed in more detail in the upcoming sections.

Table 3. Attempted NetHunter services.

Service	Attempted
SSH	YES
APACHE2	YES
POSTGRESQL	NO
DNSMASQ	YES *
Enable HID + MTP + ADB for Windows	NO
Update Kale Metapackages	NO
Launch Wifite	YES
Start wlan0 in monitor mode	YES
Stop wlan0 in monitor mode	YES
Start wlan1 in monitor mode	NO
MAC Changer	YES
KeX Manager	NO
USB Funciton Selector	YES
Image Mounter	NO
USB Network Tothering	NO
PowerSploit	YES
Windows CMD	YES
Powershel HTTP Payload	YES
DuckHunter HID	YES
Bad USB MITM Attack	YES *
Mana Wireless Attack	NO
Bluetooth Arsenal	NO
MITM Framework	NO
Nmap Scan	YES
Metasploit Payload Generator	YES
SearchSploit	NO
Pineapple Connector	NO
Wardriving	YES
DeAuth	YES *

Note. Attempts marked with * were partially executed.

The services in Table 3 were performed in hopes of finding evidence of attacks and activities carried out on the device. During the process, the researchers observed all configurations and changes with the aim of allowing easier search of the artifacts during the analysis.

3.4. Acquisition

The acquisition phase consisted of three different steps. Due to the unfamiliarity with the OS, the first acquisition took place immediately after rooting the device. Before the data population and after rooting the phone, a physical image of the phone was captured, establishing the baseline of the phone.

The second acquisition was performed right after the installation of the NetHunter Store, NetHunter Terminal and NetHunter applications. The purpose of this physical image was to compare the differences before and after the installation of NetHunter-related applications.

The third and last acquisition in the experiment was performed after all of the tests were conducted. This physical image was compared with the second physical image, finding the differences and attempting to match the presented data with the performed attacks.

In order to eliminate any potential software bias, the researchers ensured that the images collected were examined with at least two different software. First, the applications used to acquire the images were Cellebrite UFED 4PC [23] and the Magnet AXIOM Process [24]. The examination software used was Cellebrite UFED Physical Analyzer [23]

and Magnet AXIOM Examine [24]. All the acquisitions were performed on the same PC. The specifications of the PC included an Intel(R) Core(TM) i7-8700 CPU @ 3.20GHz processor with 16 GB RAM running Windows 10 Education 21H1.

In the following sections, we present the conducted analysis and report the findings. During the analysis, the researchers examined the last phone image taken to understand the file structure of Kali NetHunter. Additionally, images before and after the data population were analyzed using dedicated software.

4. Analysis and Findings

In this last phase of our investigation, we collected and compared the three physical images of the OnePlus 7T smartphone. The first phase included the rooted device without NetHunter Lite installed. The second physical image was taken after the installation process, and the third image was taken right after using the tools provided. As for the Android operating system process analysis, it is out of the scope of this study to analyze what is being changed by the operating system during the use of the application. Since forensic analysis is the focus in this study, we rather analyzed the data created by actions performed during user interactions. This section is divided into two subsections. The first part analyzes the findings by comparing the first and second images of the experiment. The comparison of the second and third images in which the attacks were performed is presented in the second subsection.

4.1. NetHunter Installation Analysis

The comparison of the OnePlus 7T's first and second physical image show the locations of the previously installed application. The main folders where traces of NetHunter Lite and its dependencies (NetHunter Store and NetHunter Terminal) can be found were at the following paths:

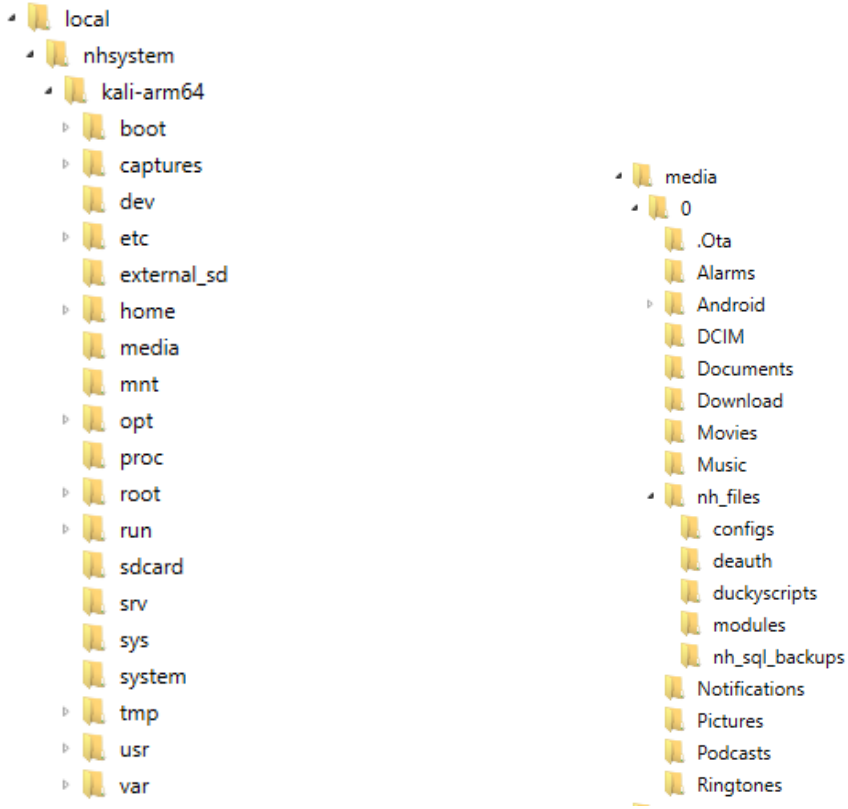
```
\data\app
\data\data
\data\local\nhsystem
\data\media\0\nh_files
\data\misc\profiles\cur\0
\data\misc\profiles\ref
\data\System\grhicsstats\163909440000
\data\system\package_cache\a415bf8295647e702d578a81b6b3801ee6f0ecad
\data\system_ce\0\launch_params
\data\user_de\0
```

Note that the above items do not include files that have been overwritten and compared with the previous image. Additionally, paths such as `\data\system_ce\0\recent_tasks`, `\data\system_ce\0\shapshots`, and `\data\backup` contained newly created content; however, it was excluded due to the irrelevance of this analysis. Moreover, all folders were examined in detail, and only readable files were analyzed.

Within the `\data\data` folder, there were three newly created folders `com.offsec.NetHunter`, `com.offsec.NetHunter.store`, and `com.offsec.nhterm`. The `com.offsec.nhterm` had two empty subfolders. Among other files, the `com.offsec.NetHunter.store` folder contained a database file 'fdroid' showing installed applications and repositories used by the application. In the `com.offsec.NetHunter` folder, there were two databases, 'NetHunterFragment' and 'KalisServicesFragment'. The first database shows different system information (e.g., Kernel Version, Busybox Version, Root Status) provided on the home screen of the NetHunter Application. The second database, 'KalisServicesFragment' shows if the services such as SSH, APACHE2, POSTGRESQL, and DNSMASQ are running. Inside the parent folder (`\data\data\com.offsec.NetHunter`), the folder 'files' contained scripts and files needed to start, stop, modify, and run the application. The scripts were divided per service, containing the paths and commands for the execution.

The next folder containing information is `\data\data\local\nhsystem\kali-arm64`. As presented in Figure 2a, this folder holds data for operations such as boot, etc, media, run, system, temp files, root. The capture folder contains any relevant files regarding Kismet, Mana, Nmap, WiFite, and other services provided by NetHunter Lite.

The `\data\media\0\nh_files` folder has five subfolders, as shown in Figure 2b. The folders contain configuration files for DNSMASQ, dhscp, dnsspoof, hid-cmd, and a file for the DeAuth service. Additionally, the folder contained duckyscripts, Python modules, and one backup file of the database.



(**a**) (**b**)

Figure 2. Partial folder hierarchy of Kali NetHunter Lite. (**a**) Local folder. (**b**) Media folder.

Other folders worth mentioning are `\data\system_ce\0\launch_params`, showing the launch parameters of the application, and `\data\user_de\0\com.offsec.NetHunter`, containing two cache files that are not readable. Folder `\data\misc\profiles` contained traces of the NetHunter application but did not present any valuable information.

4.2. NetHunter Lite Service Analysis

In this section, we analyze and compare the events that happened on the OnePlus 7T smartphone before and after the NetHunter Lite services were used. Given that the previous phase of the analysis has shown the main folders and paths of the artifacts stored, this section will look at any modified files inside those locations but it will not be limited to them. For the second go-around examination, only folders related to the NetHunter Lite application, and not the Store or Terminal, are examined.

When comparing the last physical image of the experiment with the image taken before using the services, the `\data\data` folder had created new files. The `\data\data\com.offsec.NetHunter\databases` folder had three additionally modified/created databases, 'CustomCommandsFragment', 'SearchSploit', and 'USBArmoryFragment'. In a different folder `\data\data\com.offsec.NetHunter\shared_prefs`, additional files were created. These two new files were named 'com.offsec.NetHunter.xml' and 'embryo.xml'.

The two files show a change of preferences on the chroot, language index, selected interface, and many more variables. Other files in other folders such as `\data\data\com.offsec.NetHunter\scripts` were not modified or changed.

In the folder `data\local\nhsystem\kali-arm64`, five different databases were created. These databases relate to the Kismet service that is part of the 'Wardriving' application within the NetHunter Lite Application. Out of the five databases, 'Kismet-20211211-16-11-13-1.Kismet' (see Figure 3) showed all of the scanned networks during the attack as well as the parameters such as source MAC address, destination MAC address, frequency, packet length, signal strength, etc.

Moreover, the `data\local\nhsystem\kali-arm64\root\.Kismet\Kismet_httpd.conf` file revealed the password and the username (see Figure 4) previously set for the webserver during initialization of the 'Wardriving' application.

sourcemac	destmac	frequen	signal
00:00:00:00:00:00	00:00:00:00:00:00	0	-96
00:00:00:00:00:00	00:00:00:00:00:00	0	-96
60:31:97:03:E4:D8	60:31:97:03:E4:D8	2412000	-96
60:31:97:03:E4:D8	60:31:97:03:E4:D8	2412000	-96
00:00:00:00:00:00	00:00:00:00:00:00	0	-96
60:31:97:03:E4:D8	60:31:97:03:E4:D8	2412000	-96
D0:AB:D5:CB:93:3C	D0:AB:D5:CB:93:3C	2412000	-96
D0:AB:D5:CB:93:3C	60:31:97:03:E4:D8	2412000	-96

Figure 3. Partial database from Kismet showing Wardriving logs.

PREVIEW

```
httpd_password=kismet
httpd_username=nethunter-kismet
```

Figure 4. Kismet username and password found in plain text.

Another interesting observation found was a file (`data\local\nhsystem\kali-arm64\root\.zsh_history`) showing the IP (Internet Protocol) addresses configured for the Nmap application, commands for the MITM Framework, HID Attacks configuration, and Kismet commands. Although Figure 5 shows only a segment of the file (`.zsh_history`), this section presents valuable information such as related commands and data. During the process of performing the activities, 'Bluetooth Arsenal' was not operational; however, despite the difficulties, the files were created in the `\data\local\nhsystem\kali-arm64\root\bt_audit` folder.

```
mitmf  -i wlan0 --jskeylogger
mitmf  -i wlan0 --responder --analyze
mitmf  -i wlan0 --ferretng
sudo mitmf
cd /sdcard/; msfpc asp 10.10.0.101 443 msf reverse  staged tcp
nmap  10.10.0.0-150
nmap  10.10.0.0-10.10.0.150
/usr/bin/start-kismet
kismet wlan0
kismet source=wlan0
kismet -h
kismet -f
kismet -c wlan0
/usr/bin/start-kismet
kismet -c wlan0
echo Press Crtl+C to stop! && mdk3 wlan0mon d -c
echo Press Crtl+C to stop! && mdk3 wlan1mon d -c 1
echo Press Crtl+C to stop! && mdk3 wlan0mon d -c 1
echo Press Crtl+C to stop! && mdk3 wlan0 d -c 1
ip link set wlan1 down && iw wlan1 set monitor control && ip link set wlan1 up;exit
```

Figure 5. Partial logs from .zsh_history file.

The var (`\data\local\nhsystem\kali-arm64\var`) folder revealed the files that were created in the `\www\html` folder during the 'HID Attacks'. The information found pertains to the 'PowerSploit' payload that was inserted during the attack, revealing the IP address of the local host, port, payload type, and URL of the payload. Additionally, in the `\data\local\nhsystem\kali-arm64\var\log` folder, the logs show various information on the usage of the applications and their patterns. For example, Figure 6 shows the `data\local\nhsystem\kali-arm64\rvar\log\dnsmasq.log` file and its contents. As can be seen in Figure 6, the log provides times of service usage, IP addresses, DNS (Domain Name System) names, and other capabilities needed for the successful service execution. Files in the `data\local\nhsystem\kali-arm64\etc` folder were modified but no significant information was found.

```
Dec 11 14:50:10 dnsmasq[11445]: started, version 2.85 cachesize 150
Dec 11 14:50:10 dnsmasq[11445]: compile time options: IPv6 GNU-getopt DBus
no-UBus i18n IDN2 DHCP DHCPv6 no-Lua TFTP conntrack ipset auth cryptohash
DNSSEC loop-detect inotify dumpfile
Dec 11 14:50:10 dnsmasq[11445]: warning: interface wlan1 does not currently exist
Dec 11 14:50:10 dnsmasq-dhcp[11445]: DHCP, IP range 10.0.0.10 -- 10.0.0.250,
lease time 12h
Dec 11 14:50:10 dnsmasq[11445]: reading /etc/resolv.conf
Dec 11 14:50:10 dnsmasq[11445]: using nameserver 10.10.0.1#53
Dec 11 14:50:10 dnsmasq[11445]: using nameserver 8.8.8.8#53
Dec 11 14:50:10 dnsmasq[11445]: using nameserver 208.67.222.222#53
Dec 11 14:50:10 dnsmasq[11445]: using nameserver 208.67.220.220#53
Dec 11 14:50:10 dnsmasq[11445]: read /etc/hosts - 2 addresses
Dec 11 14:55:35 dnsmasq[11445]: exiting on receipt of SIGTERM
```

Figure 6. Partial DNSMASQ.log file.

Unlike in the previous analysis, the `\data\media\0\nh_files` folder contained relevant information; more specifically, it contained information about the 'whitelist' file used during the DeAuth process and the updated 'ducky' scripts previously saved. Furthermore, the folder contained databases of 'FragmentKaliServices' and 'FragmentUSBArsenal.

The 'FragmentedUSBArsenal' database contained the IP address, gateway, subnet mask, and upstream interface previously placed by the researchers.

5. Discussion

Analyzing a new operating system is undoubtedly rewarding; however, no research comes without limitations and shortcomings. Our research was no exception to that. The most significant limitation of the research is that the Kali NetHunter is still in development and the flexibility in terms of smartphones that it is available on is restricted. Additionally, ensuring that the smartphone will be capable of performing most of the tasks narrowed down the scope of the available devices. Moreover, the inability of digital forensics software to fully read the file system was another hump on the road. While some applications were running flawlessly, with other ones we were not able to say for certain, which we will talk more about in the next paragraphs.

Throughout this research, we experienced difficulties with rooting the OnePlus 7T smartphone. The difficulties faced were due to the smartphone being proprietary to the T-Mobile provider and North America. Once the phone was converted to Global ROM, it was much easier to root it and install the necessary software. This is important to note, as the software that analyzes the phone can default to the original OS and corrupt the final image.

As mentioned previously, the researchers wanted to eliminate any potential software bias throughout the experiment, which is why Cleebrite UFED 4PC, Cellebrite UFED Physical Analyzer, Magnet AXIOM Process, and Magnet AXIOM Examine were used. While both the Magnet AXIOM Examine and the Cellebrite UFED Physical Analyzer showed the same results, the presentation of the results in the Magnet AXIOM Examine was straightforward and clearer to follow.

The methodology presented in this research was designed to explore an unknown platform, Kali NetHunter. The difference between the images was distinctive enough such that an examiner could find the differences while learning the new application. Comparing a system in three different stages ensures consistency and limits errors. The process can be replicated with many more steps, and the steps can be built upon as long as there is no contamination of the data between the phases.

Unfortunately, not all attacks were successful. Some of the attacks (e.g., KeX Manager, SearchSploit, and MITM Framework) required additional dependencies installed, which was out of the scope of the project. Other services such as Bluetooth Arsenal, did not work due to the NetHunter Lite application not recognizing the interfaces or mismatching them. Moreover, some of the services were possibly working but there was no indication if the tool was performing optimally or not. Overall, there were enough services running for the study to be successful.

The most important part of this experiment was the findings on the attacks performed that can aid mobile forensic investigators. As seen in the previous section, traces of the services utilized by the NetHunter Lite application are visible in numerous folders. Information such as IP address, DNS address, configuration files, logs, commands, and many more provide mobile forensic investigators with a road map to a better understanding of the application used.

6. Conclusions and Future Work

Mobile devices, especially smartphones, have become more diverse and powerful over the years. Diversity helps users tailor their needs and preferences when choosing a device. This diversity is not necessarily the best when it comes to digital forensics and, more specifically, mobile forensic investigations. Rapid releases and frequent updates of new devices and operating systems are primary challenges. As presented in this paper, Kali NetHunter Lite is among those operating systems that are challenging to forensically analyze due to their capabilities that might be used for malicious purposes. To aid in the research gap, this study focused on providing a detailed analysis of the application once

it was installed on the smartphone. Our results show that use of Kali NetHunter Lite on Android device creates forensically relevant artifacts that can be recovered at a certain rate and to a certain degree. Additional analysis was performed after using the services provided by the applications. Once the separate analysis was completed, a comparison of the two physical images showed the locations of the files that were stored after the tools were utilized.

Future research can be targeted to the specific services of the Kali NetHunter Lite application. Additionally, expanding the scope, installing the necessary dependencies for the unperformed services, and completing those that were not fully executed could be the focus of future work. In the case of continuation, additional devices might be needed. Separate research can be conducted on the affected devices showing the interaction of the NetHunter Lite application from a different angle. The diverse nature of Kali NetHunter allows for many different research approaches.

Author Contributions: Conceptualization, M.S. and U.K.; data curation, M.S.; formal analysis, M.S.; investigation, M.S.; methodology, M.S. and U.K.; project administration, M.S. and U.K.; resources, M.S. and U.K.; supervision, U.K.; validation, M.S.; visualization, M.S. and U.K.; writing—original draft, M.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Forecast Number of Mobile Devices Worldwide from 2020 to 2025 (in Billions)*. Available online: https://www.statista.com/statistics/245501/multiple-mobile-device-ownership-worldwide/ (accessed on 6 February 2022).
2. Stats, G. Mobile Operating System Market. Available online: https://gs.statcounter.com/os-market-share/mobile/worldwide/#monthly-200901-202203 (accessed on 7 February 2022).
3. O'Dea. Market Share of Mobile Operating Systems Worldwide 2012–2021. Available online: https://www.statista.com/statistics/272698/global-market-share-held-by-mobile-operating-systems-since-2009/ (accessed on 19 April 2022).
4. Kali Linux. Available online: https://www.kali.org/ (accessed on 19 April 2022).
5. Get Kali Mobile | Kali Linux. Available online: https://www.kali.org/get-kali/#kali-mobile (accessed on 19 April 2022).
6. Moreb, M. Mobile Forensic Investigation for WhatsApp. In *Practical Forensic Analysis of Artifacts on iOS and Android Devices*; Springer: Berlin/Heidelberg, Germany, 2022; pp. 281–328.
7. Delija, D.; Sudec, D.; Sirovatka, G.; Žagar, M. How to do a forensic analysis of Android 11 artifacts. In Proceedings of the 2022 45th Jubilee International Convention on Information, Communication and Electronic Technology (MIPRO), Opatija, Croatia, 28–31 March 2022; pp. 1042–1047.
8. Tzvetanov, K.; Karabiyik, U. A first look at forensic analysis of sailfishos. *Comput. Secur.* **2020**, *99*, 102054. [CrossRef]
9. Salamh, F.E.; Mirza, M.M.; Hutchinson, S.; Yoon, Y.H.; Karabiyik, U. What's on the Horizon? An In-Depth Forensic Analysis of Android and iOS Applications. *IEEE Access* **2021**, *9*, 99421–99454. [CrossRef]
10. Keim, Y.; Yoon, Y.H.; Karabiyik, U. Digital Forensics Analysis of Ubuntu Touch on PinePhone. *Electronics* **2021**, *10*, 343. [CrossRef]
11. Asaad, R.R. Penetration Testing: Wireless Network Attacks Method on Kali Linux OS. *Acad. J. Nawroz Univ.* **2021**, *10*, 7. [CrossRef]
12. Carthy, L.; Øvensen, E.; Little, R.; Sutherland, I.; Read, H. Committing the perfect crime: A teaching perspective. In Proceedings of the European Conference on Information Warfare and Security, ECCWS, Oslo, Norway, 28–29 June 2018; pp. 87–95.
13. Zaabi, K.A. Android device hacking tricks and countermeasures. In Proceedings of the 2016 IEEE International Conference on Cybercrime and Computer Forensic, ICCCF 2016, Vancouver, BC, Canada, 12–14 June 2016 . [CrossRef]
14. Shahriar, H.; Talukder, M.A.I.; Islam, M.S. An Exploratory Analysis of Mobile Security Tools. 2019. Available online: https://www.semanticscholar.org/paper/An-Exploratory-Analysis-of-Mobile-Security-Tools-Shahriar-Talukder/e1e9512e469051d910c17e9c330bfbd51c7ab13f (accessed on 10 August 2022).
15. Sharma, K.; Gupta, B.B. Taxonomy of distributed denial of service (DDoS) attacks and defense mechanisms in present era of smartphone devices. *Int. J. E-Serv. Mob. Appl. (IJESMA)* **2018**, *10*, 58–74. [CrossRef]
16. Dondyk, E.; Rivera, L.; Zou, C.C. Wi–Fi access denial of service attack to smartphones. *Int. J. Secur. Netw.* **2013**, *8*, 117–129. [CrossRef]
17. Farina, P.; Cambiaso, E.; Papaleo, G.; Aiello, M. Understanding DDoS Attacks from Mobile Devices. In Proceedings of the 2015 3rd International Conference on Future Internet of Things and Cloud, Rome, Italy, 24–26 August 2015; pp. 614–619. [CrossRef]
18. Cambiaso, E.; Papaleo, G.; Aiello, M. SlowDroid: Turning a Smartphone into a Mobile Attack Vector. In Proceedings of the 2014 International Conference on Future Internet of Things and Cloud, Barcelona, Spain, 27–29 August 2014; pp. 405–410. [CrossRef]

19. Barbhuiya, S.; Kilpatrick, P.; Nikolopoulos, D.S. DroidLight: Lightweight anomaly-based intrusion detection system for smartphone devices. In Proceedings of the 21st International Conference on Distributed Computing and Networking, Kolkata, India, 4–7 January 2020; pp. 1–10.
20. Chernyshev, M.; Hannay, P. Security assessment of iot devices: The case of two smart TVS. In Proceedings of the 13th Australian Digital Forensics Conference, Perth, Australia, 30 November–2 December 2015; pp. 85–94. [CrossRef]
21. Moreb, M. Introduction to Mobile Forensic Analysis. In *Practical Forensic Analysis of Artifacts on iOS and Android Devices*; Springer: Berlin/Heidelberg, Germany, 2022; pp. 1–36.
22. Hummert, C.; Pawlaszczyk, D. *Mobile Forensics—The File Format Handbook: Common File Formats and File Systems Used in Mobile Devices*; Springer: Berlin/Heidelberg, Germany, 2022 .
23. Cellebrite. Home—Cellebrite | Digital Intelligence for a Safer World. Available online: https://www.cellebrite.com/en/home/ (accessed on 19 April 2022).
24. Forensic, M. Magnet AXIOM—Digital Investigation Platform. Available online: https://www.magnetforensics.com/products/magnet-axiom (accessed on 19 April 2022).