**KF6000 – Fundamentals of Digital Forensics Investigations**

**Evaluation of Windows phone mobile forensics using Cellebrite and Windows Phone Internals.**

**Author – UNN\_W18008993**

**Completed – 11/04/2021.**

***Executive Summary***

This report discusses Cellebrite UFED and Windows Phone Internals which can be used for forensic analysis of devices running Windows phone OS and compares them using a Nokia Lumia 930 and 630. It discusses the lack of support from a forensic perspective for all Windows phone OS’s and the security features and OS architecture which make it difficult for software to acquire data off a device without using JTAG methods. While Windows Phone Internals was very hardware dependant on its success rate in placing a device into mass storage mode and requiring a lot of preparation of the file permissions, Cellebrite UFED was unable to access any data outside of the public directory and thus was only able to extract user media files. The recommendation for forensic analysis of devices running Windows phone OS is to extract using JTAG, due to the unreliability of Windows Phone Internals caused by hardware compatibility issues, and Cellebrite’s very small list of supported devices with Physical extraction.

***Background***

There is an increasing need for tools and acquisition techniques for use on mobile devices. Both iOS and Android have most of the market when it comes to mobile operating systems and both have been provided enough support for forensic tools from 3rd party companies such as Axiom and Cellebrite, a rather neglected OS in terms of mobile forensics however is Windows Phone OS.

Windows phone OS has 4 variants. Windows phone 7, Windows phone 8, 8.1 and 10. Windows 10 mobile, the latest mobile OS by Microsoft has been out of support since December 10, 2019 [1]. Jithin, Satheesh, and Jinu discuss the architecture of Window phone 8 OS and compares it to previous versions of Windows phone mobile [2]. Windows phone 7 differs to other versions of Windows’s phone, notably Windows phone 8 in that Windows phone 7, like Windows mobile is based on Windows CE whereas Windows phone 8 (and succeeding windows phone operating systems) are based on Microsoft’s Windows NT Kernel, the same kernel used in Microsoft’s modern desktop operating systems. There are certain restrictions to the mobile kernel however and is described as a “Hybrid kernel”. Jithin, Satheesh and Jinu also provide and discuss the layers of the hybrid kernel. This contains 7 layers and 4 groups of communication/user access. Group 1 consists of File and Data, Apps, and User Partition. The layers in this group have access and can communicate with each other. Developers and the device user can access these 3 layers however users cannot access app files and its metadata, and separate apps cannot communicate with each other. Group 2 consists of the OS partition, and Internal Storage, Group 3 consists of the Windows NT kernel, and group 4 consists of the UEFI Firmware/Hardware. All access to groups other than group 1 is forbidden. This setup of the “hybrid kernel” theoretically causes major issues to forensic software developers, due to the minimal access granted by default, resulting in minimal data acquisition. Through identifying the methods of both logical and physical extraction methods with Windows Phone the paper concluded that JTAG (Joint Test Action Group) extraction methods were all that is available to devices running Windows Phone OS.

JTAG (Joint Test Action Group) is described in a study conducted by Murphy, Leong, Gaffney, Punja, Gibb and McGarry as a forensic acquisition method which involves connecting to a TaP (Standard Test Access Port) to instruct the phones processor to provide all data stored in memory chips in a raw format. This procedure is conducted using specialised software and equipment [3]. This study further backs the claim of Jithin, Satheesh and Jinu in JTAG being the only viable method of extracting data off a Windows phone by providing a case scenario. This case scenario involves the use of a Nokia Lumia 520 running Windows phone 8 which was believed to have been used in a bank robbery. Multiple challenges were faced in identifying the best way to acquire data off the device. The difficulties faced were due to a passcode being present on the device. This is an issue with phones running Windows phone 8 OS due to Microsoft’s Bitlocker, an encryption process which involves the encryption of all data on the device using a 128-bit encryption key following AES (Advanced Encryption Standards). This encryption is also protected by TMP (Trust Platform Module) which is an anti-tamper security processor which creates and protects cryptographic keys while also signing them with a private key. This key is unable to be accessed by any software on the device [4]. Due to these security measures in place, the paper identified JTAG as the only viable method in conducting an acquisition into the Nokia Lumia 520.

JTAG being the only viable method of extraction is also backed up by the lack of forensic tools for Windows phone. Magnet forensics released a guide to analysing Windows phone artifacts on November 20th, 2014, stating “Unlike other devices, such as iOS and Android, JTAG and Chip-off acquisitions are the only methods to acquire most Windows phones” [5]. Due to this, Magnet offers 0 support for any device running Windows phone OS for any of their forensic tools such as Magnet Axiom Acquire/Examine.

Cellebrite UFED however, does support windows phone devices. Support for certain extraction methods have been added to the support list for Cellebrite UFED. An example of this can be seen with the Microsoft Lumia 650 which was given File system extraction support in November 2017, version 6.4.1 [6]. The extraction methods available to each device, however, varies from device to device.

Windows phone 10 OS (The latest Windows phone OS) utilizes the security techniques present in both Windows phone 8 and Windows phone 8.1. Window phone 8, 8.1 and 10 also has security features relating to the UEFI Bootloader which could also cause potential issues in an extraction of data. Windows phone 10 devices can be placed into a mode called “Mass storage mode” which would allow a user full access to the file system on the device. This mode however requires root access to the device and is locked down by the UEFI with Secure boot. Secure Boot is a security feature built into the UEFI of Windows phone’s to verify that the bootloader present in the device’s storage is legitimate and has a valid digital signature [7]. A potential workaround to this security method, and thus allowing forensic practitioners access to this mass storage mode (and all data present on the device) is Windows Phone Internals.

A potential forensic method which has only been considered by an article by Skulin, and Mikhaylov is Windows Phone Internals (WPI). [8]. WPI is a tool which is designed to be used to unlock the bootloader of a Windows phone device, allowing access to the file system. This can be greatly beneficial as it effectively allows a forensic practitioner to create a forensic image of the file system on the device, without having to utilise JTAG methods which would require a disassembly of the device. The article goes through the process of installing and flashing the unlocked bootloader onto a Nokia Lumia 640 (RM-1075).

Windows Phone internals utilises the following files to unlock the bootloader.

* Qualcomm Emergency Download drivers
* Windows Full Flash update (FFU) ROM image file.
* Lumia Emergency programmer files.

The Qualcomm emergency Download drivers are needed to be able to boot the phone into emergency download mode. These are found in the Windows Device Recovery tool for Microsoft/Nokia devices.

FFU ROM image file is needed to unlock the bootloader by flashing the rom image file to the device while in emergency mode. This will consist of either 1 or 2 files depending on whether the original FFU file supports the current OS version on the device. The second file will provide the needed files that the original FFU file cannot provide [9].

Lumia Emergency Programmer files are needed to access the emergency mode on the device. These work in tandem with the Lumia emergency download drivers.

***Testing methodology and findings***

A comparison of Cellebrite UFED/Physical Analyzer and Windows Phone Internals was done due to both currently being only listed supported methods of accessing data on a Windows phone, with Cellebrite being a specialized piece of software designed for the forensic analysis of mobile devices and Windows Phone Internals being designed to enable access to system files which are by default protected and inaccessible by Secure boot and the UEFI bootloader for protection against tampering.

Two devices will be used in this comparison.

* Nokia Lumia 930 (RM-1045)
* Nokia Lumia 630 (RM-976)

The Nokia Lumia 930 is tested on both Windows Phone 8.1 and 10. The 630 is only tested on 8.1 due to hardware limitations.

The following software was used in the comparison of both Cellebrite and Windows Phone Internals.

* Windows Phone Internals 2.8
* Cellebrite UFED 7.42.0.82
* Cellebrite Physical Analyzer 7.42.0.50
* Autopsy 4.16.0
* Windows Device Recovery Tool 3.14.07501
* Win32 Disk Imager 1.0

Autopsy is being utilized in this comparison to analyse the data provided by Windows Phone Internals. Windows Device Recovery Tool is utilized to recover, and factory reset any of the 2 devices and driver support. Win32 Disk Imager is to back up all data on the devices.

To identify which data was collected by each forensic method, the following actions were performed on each device.

* 5 web searches.
* 2 photos and 1 video.
* Various audio tracks.
* 4 email addresses.
* Text message sent and received.
* 1 Phone call.

While testing for email address collection is possible as it is logged, testing the collection of emails received and sent is not. This is due to Microsoft shutting down the internal services needed to log into email accounts on all Windows phone operating systems.

Both the 930 and 630 is reset to factory defaults after each forensic method is tested with the same files and actions being performed on the devices each time. The reason for this is to remove any potential artifacts or files left behind during unlocking/extraction.

*Windows Phone Internals*

Both the FFU and Lumia Emergency Programmer files are downloaded via Windows Phone internals using the built-in download menu. The programme attempts to scan for the correct files and checks them based on the device credentials and OS version.

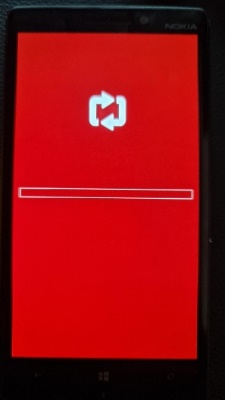


Windows phone Internals – Lumia 930

The Nokia Lumia 930 utilises the following files.

* RM1045\_02540.00019.15236.54005\_RETAIL\_prod\_signed\_1010\_027134\_ORG-GB.ffu
* MPRG8974\_fh.ede
* RM1085\_1078.0053.10586.13169.035242\_retail\_prod\_signed.ffu

The process of rebooting into flash mode and attempting to scan for flashing profiles was attempted 7 times. Each time the scanning process never got passed attempt 4 for scanning. Each time this issue occurred the device would be permanently stuck in flashing mode and required a full reinstallation of the operating system for the device to be functional again. This was attempted on both Windows 10 and Windows 8.1 with the same result. The reason for this failing/incompatibility is unknown.

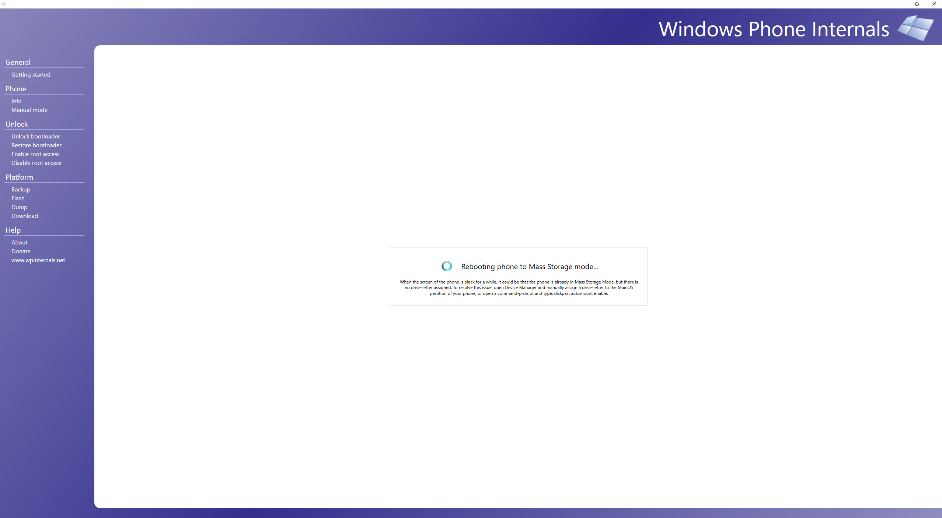


Windows phone Internals – Lumia 630

The Nokia Lumia 630 utilises the following files.

* RM976\_02040.00021.15235.50006\_RETAIL\_prod\_signed\_1115\_02D516\_000-GB.ffu
* MPRG8x26\_fh.ede
* RM976.edp

The process of rebooting into flash mode and attempting to scan for flashing profiles was attempted 6 times. 5 times resulted in the error “Failed to write pipe 4E4E4B44”. Unlike the Nokia Lumia 930, the device is not stuck in flashing mode and another attempt can be made. On the 6th attempt Windows Phone internals was able to identify and install the flashing profile and unlock the bootloader. To identify the consistency of this error, the device was reset, and the flashing process was attempted and was successful without receiving any errors.



The success of Windows Phone Internals seems to rely on the hardware as opposed to the operating system and software installed. While the Nokia Lumia 630 failed for a short period, the Nokia Lumia 930 was unsuccessful on each attempt with each attempt resulting in the device being inoperable. Success with other Windows phone devices will rely on the hardware and the success of Windows Phone Internals finding a flashing profile.

Mass storage mode

With the flashing profile successfully installed, the Nokia Lumia 630 can now be rebooted into “Mass storage mode”. This reboot must be done by using Windows Phone Internals. A Nokia Lumia 630 in mass storage mode will be identified as a Disk image on Windows 10 (desktop) with access to the file system previously hidden by secure boot, including data such as call logs, search history, SMS, contacts, and other data which may be relevant to an investigation.

*Autopsy – Analysis of data on Lumia 630.*

Permission prerequisites to analysis.

Before analysis of the file system can be performed, an issue regarding file permissions was discovered very early on into analysis. Analysis through both Autopsy and manual methods suffered critical issues in identifying key information due to insufficient permissions and access to the drive. This is caused by the set owner of these files not including authorised users on the host machine. Autopsy for example, was unable to access any information located within \USERS\, resulting in Autopsy only being able to identify media files located in OS and Program folders, saved email addresses, and deleted system files.

The owner of these directories needs to be changed with read and write permissions to access many directories located in Data. \USERS\WPCOMMSERVICES\APPDATA is an example of a directory where access would be vital. The directory Data\USERS\WPCOMMSERVICES\APPDATA includes 2 files with high forensic value. Unistore\store.vol, which stores data for SMS and contacts, and UserData\Phone, which stores all call data on the device. The ownership of these directories needs to be changed through windows command prompt (cmd.exe) as changing the owner of the file through Advanced Security Settings is insufficient.

Before file permissions were changed, the device was backed up usingWin32 Disk imager. The reason behind this is because changing the file permissions may cause issues with some applications once the device is rebooted back into the OS.

The following commands need to be utilized to gain ownership of these directories.

* “takeown /f "J:\Data" /a /r /d y”

takeown is a command which allows an administrator of a machine to acquire ownership of any file. /f is the specified directory which these changes will be made to, /a provides ownership to the administrators group, /r performs the previous command, and /d y is a suppression of a confirmation prompt which is triggered when the current user does not already have access to these files.

A picture containing graphical user interface

Description automatically generated

* “icacls "J:\Data” /t /c /grant administrators:F System:F everyone:F”

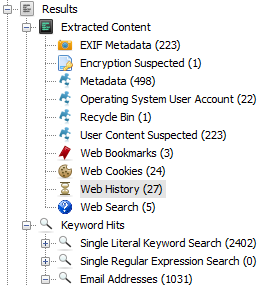
icals is a command which updates the DACL (discretionary access control list) in the specified directory, updating the DACL to confirm the administrator group as the owner of files in this directory.

With the ownership of the directory changed, Autopsy, and manual examination techniques can access all files located in “Data” and should be unhindered.

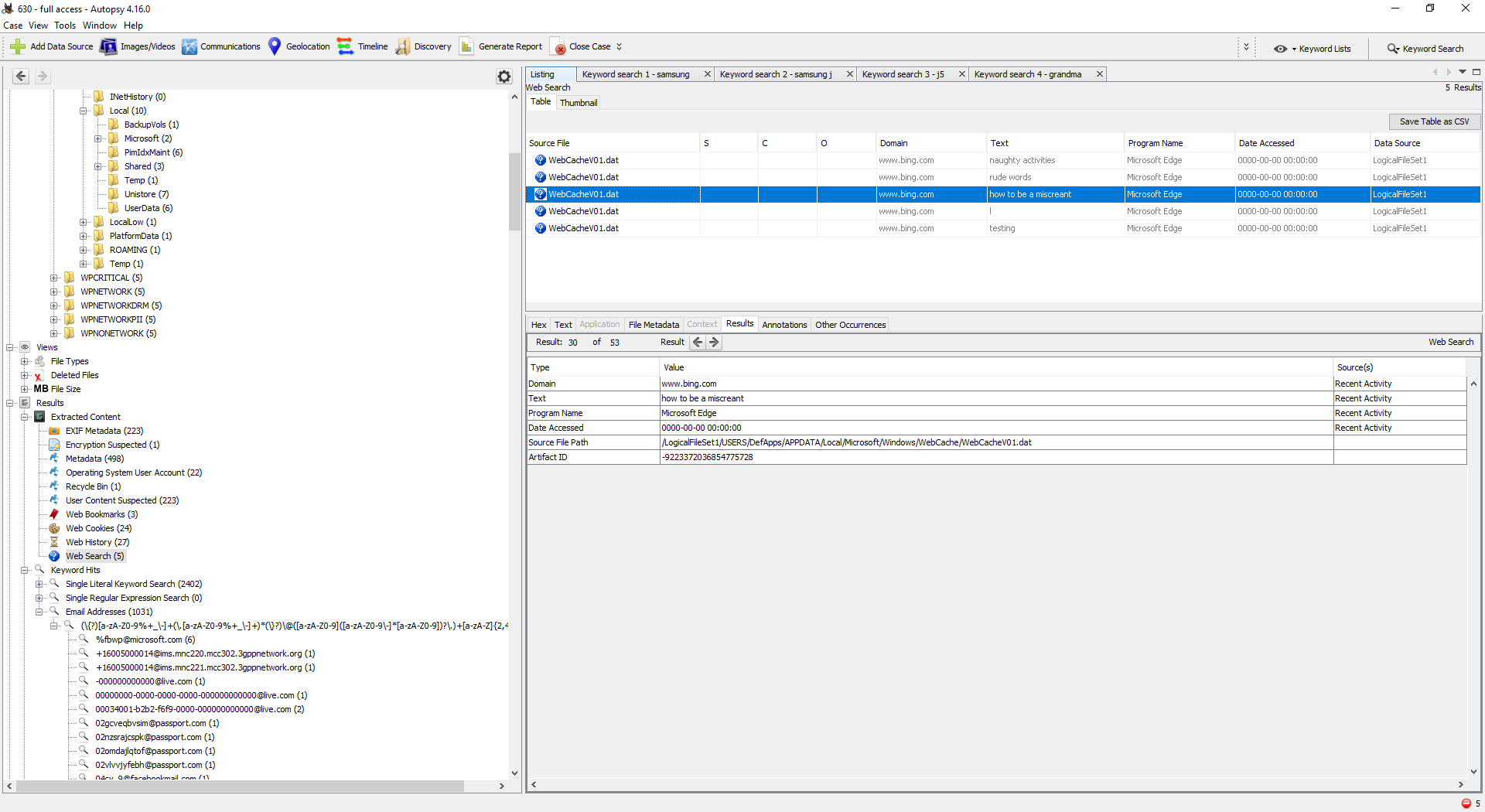
Autopsy

Once access to the file system on the Lumia 630 has been given to the administrator group, Autopsy is able to access and analyse the file system of the Nokia Lumia 630. With Autopsy, it is also important to either specify multiple data sources, including both the Disk image of the Lumia 630 as well as logical analysis of the files located in the disk volume “Data”, or for a forensic image to be created of the Nokia Lumia 630’s file system for analysis. The reason for this is due to Autopsy being unable to read the disk volume “Data” if processing the device as a disk image (live forensics). If a forensic image of the file system is performed and the analysis using Autopsy is performed via logical analysis, then the issue is not present.

Without manually searching through the Data sources, Autopsy was able to provide the following results.



All actions regarding web searches, history, and media files were identified by Autopsy and placed into the correct categories.



Data regarding call, SMS, and contacts is not displayed in results by Autopsy. This is due to a lack of support for Windows phone directories. By default, Autopsy only has support to collect SMS and call data from Android devices. This data much be identified manually.

SMS and contact data are in \USERS\WPCOMMSERVICES\APPDATA\Unistore\store.vol and can be analysed through Autopsy by analysing the text data of the file manually. Contact information can be seen on page 2 and previous received and sent texts can be seen on page 3. Call data can be found in \USERS\WPCOMMSERVICES\APPDATA\UserData\Page and can be seen on page 1.

Graphical user interface, text, application, email

Description automatically generated

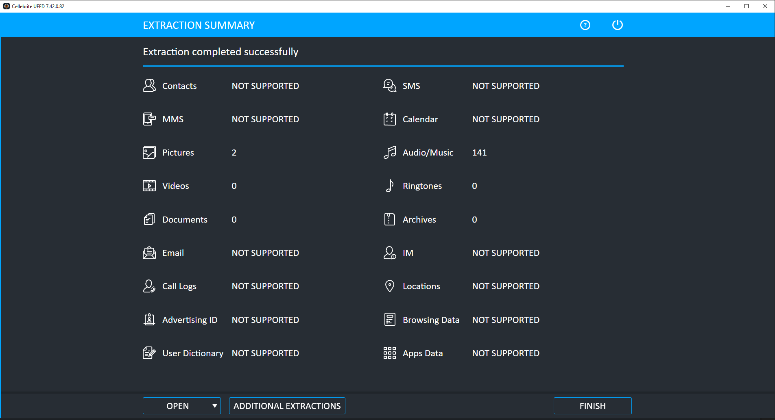
*Cellebrite analysis*

Cellebrite provides the option for us in Cellebrite UFED to analyse and extract data from mobile devices, ranging from multiple operating systems. Windows phone 7, 8, 8.1 and 10 are listed as supported and extraction methods are provided, however the extraction methods available vary depending on the device itself.

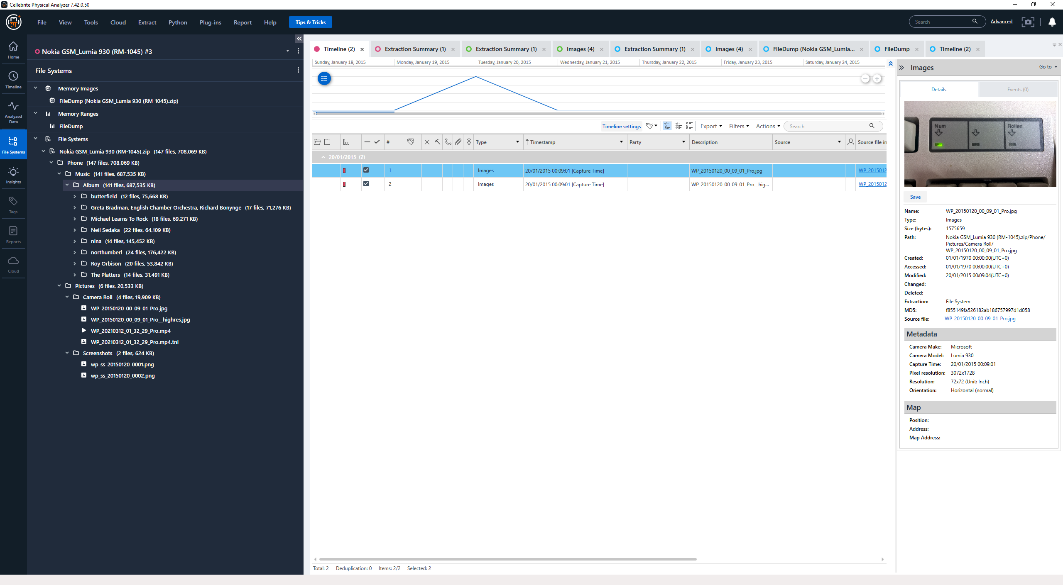
The Nokia Lumia 630 and 930, despite being listed as supported, are given very few options for extraction methods. There are 2 relevant extraction methods available for both devices.

* Logical – Attempts to extract supported file types.
* File System – Attempts to extract all allocated memory.

The physical extraction method on Cellebrite UFED is only available to a small select few of Nokia Lumia devices. An example of a device which is listed as supporting this extraction method if the Nokia Lumia 928 (RM-860). Physical extraction is an attempt to extract both allocated and unallocated data on the device including but not limited to the file system, deleted files, hidden files, and locations. Physical extractions on Nokia Lumia devices are performed by placing the device into “recovery mode” to gain access to the full file system. This method cannot however be tested due to the 930 and 630 not supporting this feature.



Logical extraction of both the Nokia 930 and 630 provided very little support for data types. The only data supported by logical extraction on these devices were media files, documents, and archives with logical extraction only identifying 2 pictures and 141 audio files. All other data types such as SMS data and Email information is not supported. File System extraction, despite being designed to identify and acquire all allocated memory, was only able to extract media files. The data acquired from Logical extraction was identical to the data acquired by file system extraction. This data is examined within Cellebrite Physical Analyzer with all examination tools enabled.



The reason for the lack of data acquired from “file system” extraction is due to Cellebrite only having access to data made available through the “public” file directory. The public file directory only allows access to media files and documents on the device. This directory is what is available by default to the user without modifications to the bootloader and placing the device into mass storage mode, as seen with Windows Phone internals. The file directory for the public file when accessing in mass storage mode is \Data\USERS\Public.

***Discussion and Conclusions***

*Findings from a practical perspective*

From a practical perspective, both methods of software extraction have their issues and complications. Cellebrite, while providing wide, although limited support to Windows phone devices, fails to provide any usable data other than that already accessible through public folders on the device, data which could be extracted by simply connecting the device to a machine running a compatible OS. This is true for all extraction methods available to both the 930 and 630. Physical extraction is a potential option on a few devices however these are few and far between and not applicable to the 930 and 630. The options Cellebrite provides to most devices is too insubstantial to be considered as a main method of extraction for Windows phone devices.

Windows Phone Internals, while providing a much wider range of support for Windows phone devices through unlocking and flashing the bootloader, this method was unreliable with both the 930 and 630, with the 630 being the only device to successfully be flashed. In addition to the data on the 930 becoming inaccessible through conventional means due to Windows Phone internals failing to flash profiles, causing the device to be stuck in emergency flash mode. The risk of accessing data through Windows phone internals for forensic acquisition is too great to be considered without prior testing of a clean, identical device first unless JTAG methods are already readily available.

Windows phone internals, while being more unreliable in terms of the success rate, is the better option of the 2 as it allows access to usable data other than those already available through the public access folder.

*Comparison to previous work*

Both our findings and previous work identified both JTAG and Chip-Off extraction as the main option for extraction. Murphy, Leong, Gaffney, Punja, Gibb and McGarry however, were the only paper to acknowledge the use of Cellebrite with Windows phones, however Cellebrite was used to examine a file system which had been extracted using JTAG and identified that as the only viable method for Windows phone devices [3]. Jithin, Satheesh and Jinu also mention Cellebrite, however they only mention it as a tool generally used for mobile forensics and do not use Cellebrite with Windows phone devices.

There are no identified previous papers which mentions Windows Phone Internals. An article by Skulin, and Mikhaylov do however make mention to Windows Phone Internals in a forensic setting and advise it as a potential alternative to JTAG and Chip-Off methods. [8]. They did not however, have any issues with compatibility with WPI, using a Nokia Lumia 640 (RM-1075). They also did not provide any details of potential changes to user permissions to certain files, as was experienced from our testing in attempting to access data on a device in mass storage mode.

*Recommended improvements to the state of digital forensics.*

Windows 10 mobile has been unsupported since January 14th, 2020. Due to this, the importance of proper software support in forensic tools for Windows phone devices is not as important as it was when the platform was in relevance. Despite this, it is recommended that efforts are made to support more Windows phone devices on forensic tools such as Cellebrite. Despite many services being shut down, Windows phone devices can still be used for illegal activity and with this data being inaccessible due to the bootloaders anti tampering, the knowledge to remove these security features and the ability to access this data is vital, especially if JTAG methods are unavailable.

***References***

1. Microsoft (2019) “Windows 10 Mobile End of Support: FAQ – Microsoft Support”.
2. Jithin, S., Satheesh Kumar, S. and Jinu Kumar, S., 2015. A novel method for windows phone forensics. Int. J. Sci. Eng. Res, 6(1).
3. Cynthia Murphy, Adrian Leong, Maggie Gaffney, Shafik G. Punja, JoAnn Gibb, Brian McGarry (2021) “Windows Phone 8 Forensic Artifacts”
4. Microsoft (April 2014) “Windows phone 8.1 Security Overview”
5. Magnet Forensics (November 20th 2014) “Analyzing Windows Phone Artifacts with IEF” ( https://web.archive.org/web/20141206082846/http://www.magnetforensics.com/analyzing-windows-phone-artifacts-with-ief/)
6. Cellebrite (November 2017) “UFED & UFED InField Version 6.4.1 Maintenance release”
7. Microsoft (October 13th 2017) “Windows 10 Mobile security guide”
8. Oleg Skulkin, Igor Mikhaylov (June 3rd 2018) “Windows Phone Physical Imaging Without JTAG and Chip-off”
9. Ehathcliff74 (2018) “Windows Phone Internals – Getting started – Required Third Party Downloads”.