A Novel Method for Windows Phone Forensics

Jithin S, Satheesh Kumar S, Jinu Kumar S V

Abstract— Mobile forensics is a branch of cyber forensics which helps in extracting evidence from mobile devices. A variety of software tools are available from different vendors for performing the acquisition and analysis of handheld devices ranging from basic phones to smart phones. From an investigator's point of view, information like call log, sms, mms, contacts, multimedia and other user related files are the important artifacts that have to be extracted and analyzed from such devices. The commercially available software tools have different capabilities in extracting these data depending on the make and model of the device under investigation. This research paper emphasizes on the forensic analysis of one of the popular smart phone operating systems named Windows Phone. Windows Phone is relatively a new smart phone operating system with the potential to become one of the major smart phone platforms in the near future. This research paper discusses about the feasibility of conducting a logical acquisition on the device and details the artifacts that can be extracted from the device.

Index Terms— Digital forensics, windwos phone, smart phones, security, cyber crimes, acquisition

———————————— ◆ ————————————

1. Introduction

Smart phones have revolutionized the daily life of people in the past decade. They keep developing by including more and more features and facilities. With the current pace of technology it is sure that Smartphones will further enhance their domination on lives. Alongside, the crimes committed using Smartphones are getting sophisticated. The cyberspace created by Smartphones is being exploited by hackers, crackers, terrorists, vandals and other sorts of cyber criminals. As new varieties of Smartphones are being introduced in the market, a systematic and classified research on smart phone forensics is inevitable.

Cell phone forensics is a branch of digital forensics which deals with the recovery of digital evidence from cell phones and other handheld devices. The term mobile devices include mobile phones, tablets, GPS devices, PDA devices etc. A lot of software tools are available in the market for conducting acquisition and analysis of mobile devices. Major mobile device forensic software tools like Cellebrite's UFED, Oxygen forensics tool, .XRY, Lantern etc have their own features in acquiring and analyzing the phones. Some of these tools support a wide range of phones, while some others are specific on particular devices or operating systems. This research paper focuses on the forensics of Windows Phone Operating system, a new operating system with many advanced security features.

Windows Phone is relatively a new smart phone operating system developed by Microsoft. It was released in September 2010. Since its inception, the popularity of the OS and Windows Phone (WP) devices are growing drastically. Hence the crimes using WP is also on the rise. Windows Phone Forensics is highly relevant on this context. Two major revisions have occurred since the release of this operating system. Windows Phone 7 and Windows Phone 8. These two versions have a lot of differences in their architecture, functionality and its implementation in the device. The internal details of the phone are explained in the later sections. The solutions presented in this paper are tested with the latest version of WP devices and the methods used are backward compatible unless mentioned.

2. Cellphone Forensics

Cell phone forensics alias mobile phone forensics is an emerging branch of digital forensics. Here the information available in the phone will be acquired and analyzed separately without using the conventional tools for disk and other storage devices. The main reason for not using the disk forensics tools is that the bit by bit copying of memory is not always possible in mobile phones. This is because the flash memory used in mobile phones is restricted from physical acquisition by the tools. Also sometimes external software agent is required to be installed in the device for acquiring the physical memory. One important issue with mobile forensics tools is that, each brand or model requires separate modules for acquisition as it is different from the other. No common data transfer protocol is available for all phones. Another difference is that contrary to media hash of disk storage devices, the hash value will be different in mobile phones if we physically hash the device second time. This is due to the reason that many a time an agent is to be installed in the device and hence no write-blocking devices can be used in mobile phones. Also a phone must be switched on and/or may be synchronized with the forensics workstation for acquisition. Hence mobile phone forensics needs to be carried out using specific software or hardware tools with procedures introduced by National Institute of Standards and Technology (NIST).

3. WINDOWS PHONE

Windows Phone is an easy-to-use smart phone OS and has made a significant progress in the global smart phone market. Two major versions have been released since its inception: Windows Phone 7 and Windows Phone 8. The latest version of this smart phone OS is Windows Phone 8.1 which was released in April 2014. Prominent manufacturers like Nokia (now Microsoft Mobile), Samsung, HTC, Dell, Hp etc. have products based on Windows Phone OS.

Windows Phone based devices have a lot of facilities and features that make these devices a significant competitor with Android, iOS, Blackberry and other smart phone operating systems. Its user interface code named metro consists of live tiles for different applications; features and functions are similar to desktop version of Windows 8. A built-in browser Internet Explorer facilitates web browsing. This browser is the miniature version of the PC based Internet Explorer with the same trident engine. All Windows Phone devices have Microsoft Office suite for handling documents and have the support of OneDrive and Office-365. OneDrive is the cloud storage facility provided by Microsoft and this is the default cloud storage for Windows Phone. User documents, photos, and other files can be synchronized with this facility. Like other smartphones, these devices support almost all multimedia formats. Apps are readily available for playing multimedia files. Windows Phone also supports games and other apps for better user experience. Windows Phone Store is the platform for distributing apps and games. The features explained above makes Windows Phone a smart and strong competitor in the market.

3.1 Architecture of Windows Phone

Windows Phone 8 shares the same core of Windows 8 PC version [3]. Unlike its predecessors Windows mobile and Windows Phone 7, which was based on Windows CE or Windows Embedded, this version of Windows has a lot of user friendly features and facilities. Though the PC and mobile phone versions share the same core, there are certain restrictions regarding the access rights and privileges. The layered representation of hybrid kernel of the Windows Phone 8 architecture is shown in Fig.1.

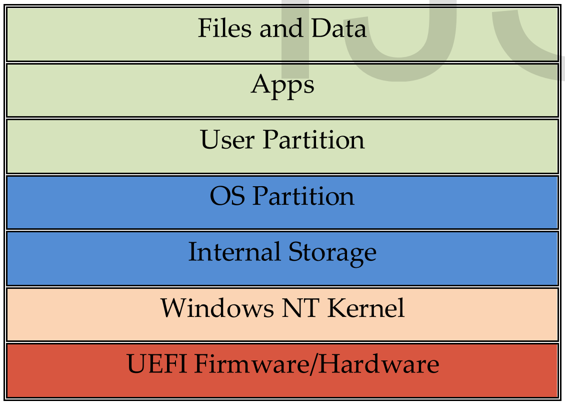


Fig.1 Windows Phone Architecture

The top three layers comprise one group where they have access to each other. The user files and data reside at the top layer. Apps form the second layer and user partitions in the third layer. The end users and developer can access the files and data layer, user partitions and the apps layer. The access to app layer simply means the usage of apps in the device and the access to the app related files and its metadata is restricted. In addition to this, the apps cannot communicate to each other. Though there are certain exceptions for this property, an agent (app) based approach for accessing device memory and related resources is not possible here because of this restriction. Microsoft calls this property as "Isolated Storage" which means the apps are compartmentalized and restricted from communication. The layers beneath user partition consists of critical components of the operating system like OS partition, Internal Storage, Windows NT Kernel, and UEFI (Unified Extensible Firmware Interface) Firmware which are again forbidden for developers and users.

4. LOGICAL ACQUISITION

Logical acquisition of any device means the extraction of highest level contents of the file system. It usually includes the files and folders that can be viewed with the help of file explorers of operating systems.

Windows Phones do not provide sufficient privileges to the end user for viewing system files or app data. The device, when connected to a Windows system, is recognized as a portable device with limited visibility of file system. It means that the users do not have access to the system files or the files associated with the apps they have installed. This happens due to the specific design of the Windows Phone Operating System architecture. The documentation about the Operating System provided by Microsoft show beyond doubt that these features are definitely intimidating remarks for investigators. The logical acquisition of Windows Phone consists of extracting contacts, calendar and other files from the device. The high security features of the phone do not allow acquiring other data such as SMS, call logs from the device.

4.1 Contacts and Calendar data acquisition

Extracting contacts from phone and establishing the link between the suspect and others involved in a crime is a major part of any investigation. The acquisition of phonebook is carried out using an agent based approach. An agent application is developed using the Windows Phone SDK. The agent is installed in the device as a Windows Phone app with the extension .xap. The app deployed on the device has the capabilities to access contacts and appointments. The contact and calendar data resides in the phone as device specific data types called Contacts[10] and Appointments[11] respectively. The underlying information like phone numbers, address, birthdays, other appointments etc. are not developer friendly data types like string, character array or integers. So a custom class was created for handling such data. This custom class consists of data members of programmer friendly data types like string, integer or character array. The contacts stored in the device specific data types are converted to string and assigned to the custom class members. The agent makes the contact and calendar information ready for sending to forensics workstation, after it executing in the phone. The steps involved in developing the app are illustrated below.

4.2 Agent implementation

The agent application is implemented in C++. The main steps involved in the implementation are explained here. First create objects of Microsoft.Phone.UserData.Contacts and Microsoft.Phone.UserData.Appointments as follows. Microsoft.Phone.UserData.Contacts contacts = new Contacts(); Microsoft.Phone.UserData.Appointments appointments = new Appointments(); Then Invoke Search async of Microsoft.Phone.UserData.Contacts class. The API required for contacts is contacts.SearchAsync
(searchterm, FilterKind.DisplayName, null) and The one for calendar is appointments.SearchAsync (start, end, max, "Appointments Test #1"). Then Search completed event will be invoked after searching has completed for contacts as well as calendar. private void contacts_SearchCompleted (object sender, ContactsSearchEventArgs e) private void appointments_SearchCompleted(object sender, AppointmentsSearchEventArgs e) are the required API calls. Contacts and calendar are get stored respectively in results of ContactsSearchEventArgs and results of AppointmentsSearchEventArgs.Results shall be transferred to a master table depending on the data. The API used are
IEnumerable<Microsoft.Phone.UserData.Contact>
PhoneBook_Master = null; and
IEnumerable<Microsoft.Phone.UserData.Appointment>
Appointments1_Master = null; the results are available at
results->System.Collections.Generic.IEnumerable
<Microsoft.Phone.UserData.Contact>
PhoneBook -> PhoneBook_Master;
results -> System.Collections.Generic.IEnumerable
<Microsoft.Phone.UserData.Appointment>
Appointments1 -> Appointments1_Master

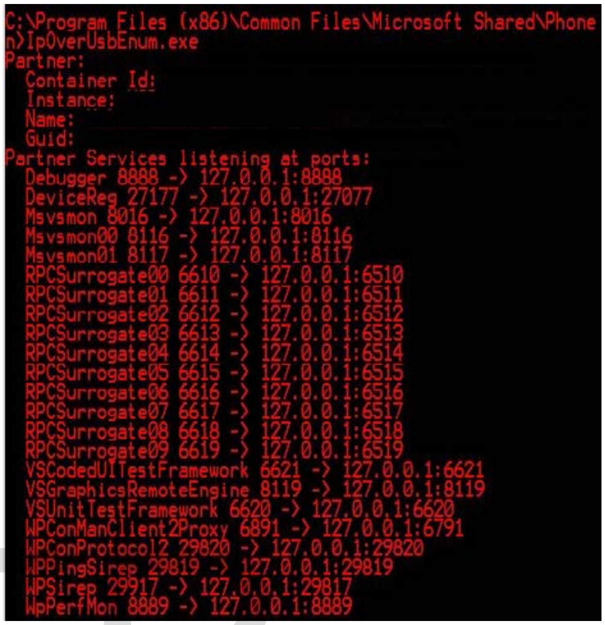
4.3 Establishing connection

Next step is to send these string data to the system connected. Windows Phone 8 and Windows Phone 7.5 support socket communication over USB and this method can be applied for such devices running on these platforms. However the first version of Windows Phone i.e. Windows Phone 7 does not support socket communication over USB. Here the acquisition of contacts and calendar information is carried out using Bluetooth communication.

In order to send the acquired data to desktop PC, first is send the custom class variables to the system. As mentioned above, socket communication can be utilized for transferring the data. The device consists of certain open ports with which a connection can be established between the phone and the workstation. These open ports can be checked using the "IpOverUsbEnum.exe" tool available with Windows Phone Software Development Kit. The sample output in command prompt is shown in Fig. 2.

As the device is a part of the local host, the IP address 127.0.0.1 can be used for communicating with the app. A

number of ports are open in the device as shown in the fig. 2. The proposed tool can make a connection with the debugger port of the device i.e. port 8888[12]. The device then establishes a connection with the local host using the IP 127.0.0.1 and the port 8888. The required socket operations for sending the string are performed using this connection. Proper synchronization between the device and the system is required for sending the data.



```
C:\Program Files (x86)\Common Files\Microsoft Shared\Phone
n>IpOverUsbEnum.exe
Partner:
  Container Id:
  Instance:
  Name:
  Guid:
Partner Services listening at ports:
  Debugger 8888 -> 127.0.0.1:8888
  DeviceReg 27177 -> 127.0.0.1:27077
  Msvsmon 8016 -> 127.0.0.1:8016
  Msvsmon00 8116 -> 127.0.0.1:8116
  Msvsmon01 8117 -> 127.0.0.1:8117
  RPCSurrogate00 6610 -> 127.0.0.1:6510
  RPCSurrogate01 6611 -> 127.0.0.1:6511
  RPCSurrogate02 6612 -> 127.0.0.1:6512
  RPCSurrogate03 6613 -> 127.0.0.1:6513
  RPCSurrogate04 6614 -> 127.0.0.1:6514
  RPCSurrogate05 6615 -> 127.0.0.1:6515
  RPCSurrogate06 6616 -> 127.0.0.1:6516
  RPCSurrogate07 6617 -> 127.0.0.1:6517
  RPCSurrogate08 6618 -> 127.0.0.1:6518
  RPCSurrogate09 6619 -> 127.0.0.1:6519
  VSCodedUITestFramework 6621 -> 127.0.0.1:6621
  VSGraphicsRemoteEngine 8119 -> 127.0.0.1:8119
  VSUnitTestFramework 6620 -> 127.0.0.1:6620
  WPConManClient2Proxy 6891 -> 127.0.0.1:6791
  WPConProtocol2 29820 -> 127.0.0.1:29820
  WPPingSirep 29819 -> 127.0.0.1:29819
  WPSirep 29917 -> 127.0.0.1:29817
  WpPerfMon 8889 -> 127.0.0.1:8889
```

Fig.2 Output of IpOverUsbEnum.exe

4.4 Deployment of agent

Windows Phone does not support a flexible deployment of apps. The agent can be deployed in two ways: app deployment to a developer unlocked device or downloading the app from Windows Store. The first method is tedious for naïve investigators. It can be done using Windows Phone Developer Registration Tool. Connect the device to the system and make sure that an active internet connection is available on both system and the device. The device can be developer unlocked by clicking on the "Register" button on the tool. The tool is a stand-alone tool, but it comes with the Windows Phone Software Development Kit.

The other method is to develop the agent and upload it to the Windows Phone Store and get it signed by Microsoft. After that, the agent will be available in the store for download. The agent can be directly installed into the device using the Windows Store app or download the agent file and copy the ***.xap*** file into the SD card of the device. The agent can be then installed using the local app installation option available in the store.

4.5 Files system acquisition

Files and folders can be acquired using the APIs[9] provided by Microsoft for Windows Portable Devices. The IPortableDevice[7] interface is an interface for C++ that provides an access to a portable device. The client interfaces are designed to be used for any WPD object. The steps involved in this phase are enumeration of devices connected to the system, content enumeration of the selected device and copying file contents of the device to the workstation system.

5. PHYSICAL ACQUISITION

Physical acquisition of a device refers to the bit by bit copying of the device memory. This process creates the exact replica of the device memory. The output of this process is usually a single image file or fragmented image files for better storage options. The disk image output obtained using ***dd*** command is an example for this. This section discuses the feasibility of physical acquisition of Windows Phone devices.

The physical acquisition of Windows Phone is almost impossible using software. This is because of the tight security features and stringent policies that Microsoft implemented on their smart phone OS. Though these features are implemented with the best intentions of protecting their users' privacy, they pose a serious threat to the users as well as forensic investigators and law enforcing agencies by enabling the criminals to use the phone for committing crimes and tampering evidence without being caught. The following section briefly explains the various security features and policies of Windows Phone that prevents the forensics analyst from physical acquisition.

5.1 Hardware Complexity

The configuration of Windows Phone is similar to other smartphones. But in addition to normal features, it has some other security features. First of all, the entire device is encrypted using the BitLocker; the same technology found in Windows based desktops and laptops. The encryption uses 128-bit AES algorithm which is a good choice for security. And the encryption key used is protected by TPM (Trusted Platform Module). TPM based crypto processor is a part of Windows Phone hardware which makes the encryption key available only for trusted boot components. This helps to secure data and ensures device integrity.

Another hardware feature that ensures the device integrity is UEFI based booting. UEFI is a replacement for BIOS based devices which have a lot of advanced features. One of them is security. This feature prevents the jail breaking of the device and unauthorized execution of program codes. It is done by a verification of the digital signature which is burned into the firmware of the device. This secure booting prevents any boot loader or custom ROM methods for getting more privileges in the device. Because of the above cited security measures, the physical acquisition of WP is still not materialized.

6. CONCLUSION AND FUTURE WORK

Windows Phones are relatively new and Windows Phone forensics is still in its infancy. As of now, no tools are available for the complete analysis of these devices. It is because of the stringent security features implemented in the device. This paper explains how to acquire the extractable logical contents from Windows Phones. The method for physical acquisition of Windows Phone is not available right now. The possibilities for physical acquisition of the device are a major scope for future work. The options like chip off and JTAG extractions are may be used for Windows Phones also. The security features like the pattern lock, PIN, password etc are also poses challenge to the forensics community.

ACKNOWLEDGMENT

We would like to thank Mr. Thomas K L, Associate Director, Resource Centre for Cyber Forensics (RCCF), Centre for Development of Advanced computing (CDAC), Trivandrum, for his valuable suggestions and support. This work was done at the RCCF, CDAC, Trivandrum, India.

REFERENCES

[1] Andrew Whitechapel, Sean McKenna, "Windows phone 8 development internals", Published by Microsoft Press, 2013

[2] YUNG ANH LE, "Windows Phone 7: Implications for Digital Forensic Investigators", 2012, umpublished

[3] Mark Russinowich, David A. Solomon," Windows internals", Microsoft Press, 2012.

[4] George Grispos, Tim Storer, William Bradley Glisson, "A comparison of forensic evidence recovery techniques for a windows mobile smart phone", Published in Journal Digital Investigation, July 2011.

[5] Windows Phone Architecture (11 October 2014), Available: https://dev.windowsphone.com/enus/OEM/docs/Getting_Started/Windows_Phone_architecture_overview.

[6] Channel9 resources (5 June 2014), Available: http://channel9.msdn.com

[7] Enumerating Devices (20 October 2014), Available: http://msdn.microsoft.com/enus/library/windows/desktop/dd319331(v=vs.85).aspx

[8] XDA Forums (5 November 2014), Available : http://www.xda-developers.com

[9] COM Component Object Model Technologies (15 September 2014), Available: https://www.microsoft.com/com/default.mspx

[10] Contact class (20 October 2014), Available: http://msdn.microsoft.com/library/windows/apps/br224849

[11] Appointments class (11 October 2014), Available: http://msdn.microsoft.com/enus/library/windows/apps/microsoft.phone.userdata.appointments(v=vs.105).aspx

[12] StreamSocketListener Class (11 October 2014) Available: http://msdn.microsoft.com/enus/library/ie/windows.networking.sockets.streamsocketlistener