An Agent based Approach using Symbian Phone Forensics

Neha V. Londase
B.E. Final Year
Department of Computer Science & Engineering
J.D.I.E.T., Yavatmal

Radhika K. Rathi
B.E. Final Year
Department of Computer Science & Engineering
J.D.I.E.T., Yavatmal

Rhucha D. Gangamwar
B.E. Final Year
Department of Computer Science & Engineering
J.D.I.E.T., Yavatmal

ABSTRACT

Smart phones like the older mobile phones are fast becoming a life style choice. These sleek devices with large amount of personal information in them make smart phone forensics, a key component in any criminal investigation. It presents a contrast between hardware and software approaches and highlights the key advantage of the software approach i.e. the speed at which actionable data can be made available with less technical knowhow. Moreover, the proposed plug-in based agent development provides an extensible framework to handle customizations that will matchup with each unique nuance of phone platform and model. The main focus is the simplicity and extensibility of proposed approach but at the same time the paper does warn about the possible impact to device memory and contrasts with other alternatives.

Keywords

Cyber forensics, law enforcement, mobile computing, security.

1. INTRODUCTION

The use of mobile phones has grown in recent years. The added functionality of smart phones includes Personal Information Management, Internet browsing and Multimedia capabilities. Smart phone will also run a smart phone operating system such as Symbian. These powerful devices in the wrong hands will be equally disruptive. In this changing background forensic analysis of phones has become even more challenging all when the law enforcement agencies need to process a wide range of handsets quickly and get all available information to the investigating officer.

Analysis of such devices having a power of laptop or a notebook computer is a major agenda before the forensics community. The law enforcement agencies require sophisticated software as well as hardware tools for the proper analysis of digital evidence to bring the culprit before the court of law. A mobile phone can potentially contain a large amount of information related to the user's actions, determined by their communication patterns, and information such as images, video and audio recordings. As such, the information stored in a mobile phone may be important in proving or disproving theories and allegations. The Fig.1.1 shows the current penetrations of mobile phones in relation to world population and how mobile phones usage stacks up in comparison with other technologies.

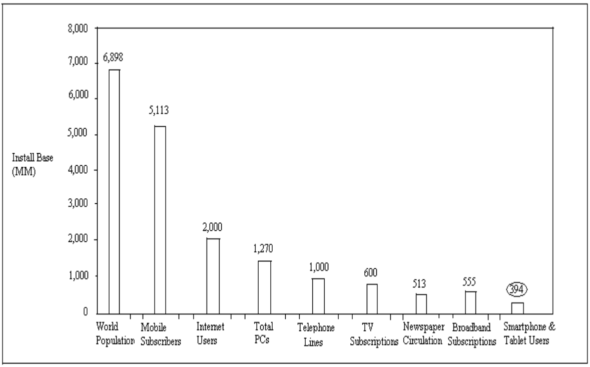


Fig1: Tablet and Smartphone Users Vs. Othe Markets

2. OVERVIEW OF SYMBIAN OPERATING SYSTEM

Symbian OS [15], [8] is an advanced operating system for mobile devices. Symbian OS certainly aims at unequaled robustness, making strong guarantees about the integrity and security of user data and the ability of the system to run without failures. Smart phones are so named because they run fully-featured operating systems and utilize the features of desktop computers. Symbian OS is designed so that it can be the basis of a wide variety of smart phones from several different manufacturers.

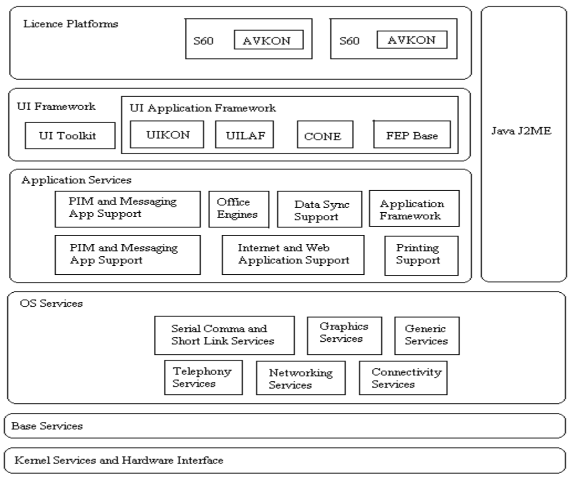


Fig 2: Symbian OS architecture

- **UI Framework Layer**
  The UI Framework layer is the topmost layer of Symbian OS. The UI Framework layer is the foundation for building customized user interfaces on top of Symbian OS and is the immediate interface between Symbian OS and the variant UI layer. The UI Framework layer provides the frameworks which custom user interfaces extend, the class hierarchies from which controls specific to the user interface are derived, and additional supporting components used primarily by user interfaces.
- **Application Services Layer**
  The Application Services layer provides user-interface-independent support for applications on Symbian OS. The Application Services layer provides services used by all applications but mediated by the UI Framework layer and the variant user interface above it, for example, application installation and launching, view switching, and the basic application architecture relationships.
- **OS Service Layer**
  In Symbian OS, the higher-level system services are located in the OS services layer. These services provide the specialized system-level support required by other system components and by higher layers of the system, as well as by applications. Thus, for example, graphics support, communications support including networking and telephony, and the connectivity infrastructure are all provided as OS services.
- **Base Service Layer:**
  The Base Services layer is the higher of the two layers and it contains the user-side servers, frameworks, libraries and utilities that build on the kernel layer to provide the basic operating system services. The Base Services layer includes a number of essential frameworks and libraries on which almost all higher-level services, as well as applications, have some direct or indirect dependencies. The Base Services layer extends the bare kernel into a basic software platform that provides the foundation for the remaining operating system services, and effectively encapsulates the user side of the 'base' operating system.
- **Kernel Service and Hardware Interface Layer**
  The Kernel Services and Hardware Interface layer is the lowest layer of Symbian OS. It contains the Symbian OS kernel and supporting components. These include the kernel-level components which must be customize in order to bring up a minimal build of the operating system on new hardware. The Kernel Services and Hardware Interface layer is the foundational layer of Symbian OS. It includes the kernel and all the supporting infrastructure needed to boot and run the kernel on the underlying hardware platform.
- **Java J2ME Layer**
  Symbian OS offers licensees an optional Java implementation. Java ME is subdivided into configurations, profiles and optional packages.

Java ME configuration provides a basic1 Java platform for a large class of constrained devices by defining a Java Runtime Execution (JRE) environment consisting of a Java language subset, a Java Virtual Machine (JVM) and a base set of necessary class libraries. Java is an important application environment for mobile phones, supporting a multiplatform third-party market for downloadable and installable programs, a standard environment for enterprise developers seeking to extend information systems to mobile phone users within organizations and a standard platform for mobile phone services, as well as a branding and personalization mechanism or network operators and others in the phone retail chain.

2.1 Symbian File System

On a Symbian smart phone, the file system [22],[2] can be accessed by means of the file server component, also referred to as F32, which manages every file device. It provides services to access the files; directories and drives on those file mapped devices. The file server uses the client/server framework, by receiving and processing file-related requests from multiple clients. Moreover, it is able to deal with different file system formats, such as the FAT format used for removable disks, by using components that are plugged into the file server itself. In addition, it supports a maximum of 26 drives, each identified in DOS-like convention by a different drive letter, in the range A: - Z. The main ROM drive, where the firmware resides, is always designated as "Z:". This drive holds system executables and data, which are referred to as XIP (eXecutable In Place) because they are directly launched without being loaded into RAM. Besides this, the firmware, or ROM image, is usually programmed into Flash memory, known also as EEPROM, the nonvolatile memory that can be programmed and erased electronically. The C: drive is always designated as the main user data drive, which can be mapped onto the same Flash memory chip of the firmware, whereas any removable media device is generally designated as D: or E:. It is worth mentioning that every access from a client to file server (F32) takes place via a file server session, by means of RFs server session class, which implements all the basic services to interact with the file system.[1] The information about drives and volumes, act on directories, notification about the state of files and directories can be obtain, analyze file names, verify the integrity of the file system, and finally, manage drives and file systems.

3. FORENSIC PROCESS

Forensics on a cell phone is considerably different from a personal computer. Methodology and sequence in which the phone is handled and data collected is critical [16]. Turning off the phone has the potential to alter its memory or data on the phone, but leaving the phone ON raises the possibility of new information arriving over the network and affecting the integrity. Ideally the phone should be placed in a radio shield environment and it should only be switched OFF if that's not possible. On the same lines removal of SIM card or battery from some phones could modify the contents of phone memory. The complexity of process is recognize and develop an application based on the plug-in model, which allows for extensive customization based on the phone platform, model and version. The application will identify the expected steps and walk the user though it. The following flowchart shows the steps involved in Forensic Process,

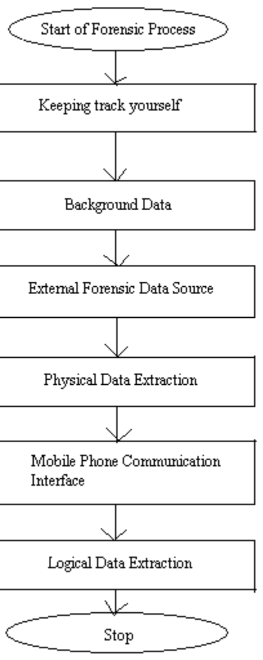

Start of Forensic Process
Keeping track yourself
Background Data
External Forensic Data Source
Physical Data Extraction
Mobile Phone Communication Interface
Logical Data Extraction
Stop

Fig3: Steps of Forensic Process

3.1 Keeping Track of Yourself

In this, each step should be accurately documented, so that there is enough information for the process to be reviewed by independent third-party. Finally, the person in charge of the investigation should maintain compliance with the governing laws. Forensic method used should minimize changes on the device, be able to retrieve the full set of data, and finally minimize user interaction with the device itself. Ideally, the full memory content of a generic embedded device should be collected, to preserve the full inner state and obtain a forensically sound acquisition. It is also recommended to keep track of approach and progress, by means of an external recording device (e.g. camera) that will maintain visual breadcrumbs.

3.2 Background Data

During the intake and processing of the phone evidence the crime scene investigator from law enforcement inputs a bunch of contextual information. This includes but not limited to where the evidence was found, the crime file details, the technician name etc. Capturing this kind of context information that can be kept with the analysis report is the first step of the forensic extraction tool.

3.3 External Forensic Data Source

There is information that can be gleaned from outside vis-à-vis the network can be as important as what is in the phone. The set of information, which can be successfully collected with this method, is related to the SIM data set, such as SMS, MMS, list of last called numbers, and the location of the subscriber. Clearly information such as photos, videos, phone book, web browser logs, audio recordings, or user's notes cannot be gathered in such a way. If external forensic data can be gathered then the request for the information from the service provider or notes regarding this are recorded by the technician.

3.4 Physical Data Extraction

Physical acquisition implies a bit-by-bit copy of the entire physical store. Physical acquisition [22] has its advantages since it allows deleted files and data remnants in unallocated memory to be analyzed. Once a bit-by-bit copy is made the extracted image need to be parsed and decoded manually. Logical extraction of data implies copying data in logical file system partition through OS frame work calls. This is a logical view not raw memory view, which has its disadvantages. Most forensic tools for cell phones and SIMs acquire more technical knowhow and training. At the minimum the technician should know how to hook up to diagnosis or debugging ports like JTAG or at the extreme level may require de-soldering the flash chip and connecting to the reader. De-soldering the flash chip is the most invasive method for the equipment so may not be the right approach in all cases. But this is ideal when the phone is damaged. If physical hardware based extraction is deemed useful the technical records those thoughts in the report log.

3.5 Mobile Phone Communication Interface

Different interfaces [3], [18] like IrDA, Bluetooth or serial cable can be used to acquire logical content. Extracting data using serial cable is the recommended option; wireless options should be used only after understanding the potential forensic issues. E.g. Bluetooth requires the wireless antenna to be switched ON and requires key entries on the handset so that it is paired with the forensic workstation and a good connection is setup all of this generates integrity concerns.

3.6 Logical Data Extraction

This is the heart of the process and relies on multiple protocol and communication methods some of the things used are AT Commands, SyncML, FBUS, MBUS, OBEX, IrMC APDU. The phone OS platform SDK provides powerful options to extract data. Because each phone has its own unique approach; the plug-in model provides an extensible option to pick and choose what works best for the phone platform and model. As you choose the phone model the correct plug-in that will do the job is called and used to extract the information. The extracted information and the final report is then run through a hashing algorithm (MD5) before saving it, to prevent tampering.

4. PLATFORM SECURITY

Platform security [23], [11], [4] on Symbian OS prevents applications from having unauthorized access to hardware, software and system or user data. The intention is to prevent malware, or even just badly written code, from compromising the operation of the phone, corrupting or stealing confidential user data, or adversely affecting the phone network. A capability grants a process the trust that it will not abuse the services related to the associated privilege. The Symbian OS kernel holds a list of capabilities for every running process and checks it before allowing a process to access a protected service. There are four different types of platform security capability, when digital signing is considered.

Table 1. Categories of Capabilities in Symbian OS

Category	**Capability**	**Brief Description**
	Local Services	Grants access to sending or receiving information through USB, IR, and point-to-point Bluetooth profiles

User Capabilities	ReadUserData	Grants read access to confidential user data
	WriteUserData	Grants write access to confidential user data
	NetworkServices	Grants access to remote services such as dialing a number or sending a text message
	UserEnvironment	Grants access to recording the user's voice and using the camera
	Location	Grants access to data about the location of the device
System Capabilities (extended capabilities)	SwEvent	Grants the right to simulate key presses, pen input, and capture such events from any program
	ProtServ	Grants the right to a server to register with a protected name
	PowerMgmt	Grants the right to kill any process in the system, to power-off unused peripherals, and to cause the mobile phone to switch its machine state
	SurroundingsDD ReadDeviceData	Grants access to logical device drivers that provide input information about the surroundings of the device Grants read access to sensitive system data
	WriteDeviceData	Grants write access to sensitive system data
System	CommDD	Grants access to communication device drivers
	DiskAdmin	Grants the right to disk administration
Capabilities (manufacturer-approved capabilities)		functions that affect more than one file or
	MultimediaDD	directory such as formatting a drive Controls access to all multimedia device drivers (sound, camera, etc.)
	NetworkControl	Grants the right to modify or access network protocol controls
	AllFiles	Grants visibility to all files in the system and extra write access to files under /private
	DRM	Grants access to alter DRM-protected content
TCB Capability	TCB	Grants access to the /sys and /resource directories in the device

There are some public areas for which no capabilities are required,

Table 2. Data Caging Capabilities

Directory	Capabilities			
	None	AllFiles	TCB	AllFiles+TCB
\resource Read	Yes	Yes	Yes	Yes
Write	No	No	Yes	Yes
\sys Read	No	Yes	No	Yes
Write	No	No	Yes	Yes
\private\<ownSID> Read	Yes	Yes	Yes	Yes
Write	Yes	Yes	Yes	Yes

\private\<otherSID> Read	No	Yes	No	Yes
Write	No	Yes	No	Yes
\<anyother> Read	Yes	Yes	Yes	Yes
Write	Yes	Yes	Yes	Yes

There are three important concepts in Symbian OS platform security: three trust tiers, capabilities, and data caging.

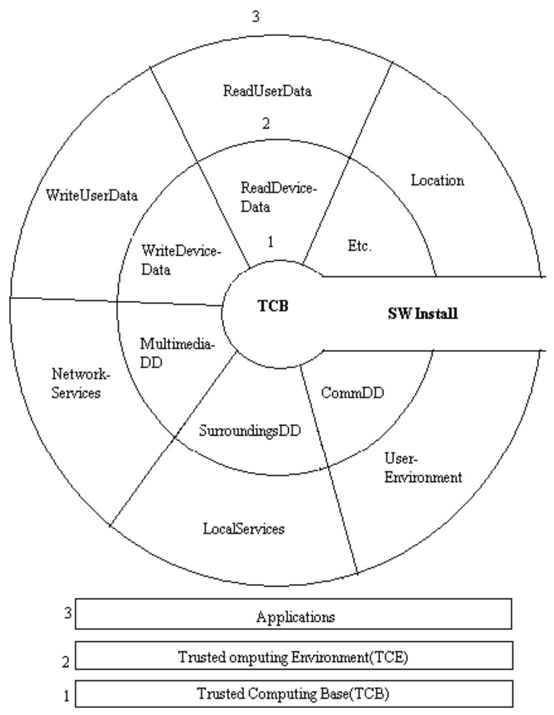


Fig4: Trusted Computing Platform: Three Trust Tiers

- **Trusted Computing Base (TCB):**
  The heart of a trusted computer system is the Trusted Computing Base (TCB), which contains all of the elements of the system responsible for supporting the security policy. TCB is the most trusted part of Symbian OS. It has three components: the operating system kernel, the file server (F32), and the software installer. The kernel has the responsibility for managing all the processes and assigning appropriate privileges to them. The file server is used to load the code for running a process. The software installer is used to install applications from files packages.
- **Trusted Computing Environment (TCE):**
  The next tier is the TCE, which is a set of different system servers running with different privileges. TCE consists of trusted software provided in the mobile phone by Symbian and others, such as the UI platform provider and the mobile device manufacturer [5]. The TCE usually implements a system server process and is less trusted, and thus it has only limited privileges to perform a defined set of functions. With this design, Symbian can ensure that failure of one server will not threaten the whole system.
- **Applications:**
  Third-party applications are the last tier in this trusted computing model. There are actually two kinds of applications: signed applications and unsigned applications. If a signed application wants to use some service provided by TCE, it must request that TCE run the service on its behalf. An unsigned application, then, is not necessarily worthless software or malware: an application should have a signature only when it is necessary.

The original policy file related to a Nokia Symbian based smart phone is illustrated as follows.

```
AllowUnsigned = false
MandatePolicies = false
MandateCodeSigningExtension = false
Oid = 1.2.3.4.5.6
Oid = 2.3.4.5.6.7
DRMEnabled = true
DRMIntent = 3
OcspMandatory = false
OcspEnabled = true
AllowGrantUserCapabilities = true
AllowOrphanedOverwrite = true
UserCapabilities = NetworkServices LocalServices
ReadUserData WriteUserData UserEnvironment
AllowPackagePropagate = true
SISCompatibleIfNoTargetDevices = false
RunWaitTimeoutSeconds = 600
AllowRunOnInstallUninstall = false
DeletePreinstalledFilesOnUninstall = true
```

It is interesting to observe that the capability set defined in the previous file is limited, and it restricts the interaction with the file system of the mobile platform. According to the standard documentation, the various parameters can appear in any order. Moreover, User Capabilities set might be changed, by adding the required capabilities such as those illustrated in the following modified version of policy file.

```
AllowUnsigned = true
MandatePolicies = false
MandateCodeSigningExtension = false
Oid = 1.2.3.4.5.6
Oid = 2.3.4.5.6.7
OcspMandatory = false
OcspEnabled = true
AllowGrantUserCapabilities = true
UserCapabilities = AllFiles DiskAdmin
NetworkServices
LocalServices ReadUserData WriteUserData
UserEnvironment MultiMediaDD NetworkControl
CommDD ReadDeviceData WriteDeviceData
SISCompatibleIfNoTargetDevices = false
AllowRunOnInstallUninstall = true
AllowPackagePropagate = true
DeletePreinstalledFilesOnUninstall = true
```

The illustrated policy file can be written directly in the original firmware of the phone and, subsequently, up loaded by means of re-flashing. As a result, the complete C: disk content might be collected, with standard self signed APIs, and thus analyzed, to extract the full set of probatory data which might be usually found on a mobile platform. This is the usual approach for other mobile platforms as well, where the primary image, the one which contains the entire set of evidence, can be obtained without any restrictions. Unfortunately, such a scenario is far from the reality, and we need to evaluate others ways to access the digital data content of the smart phone.

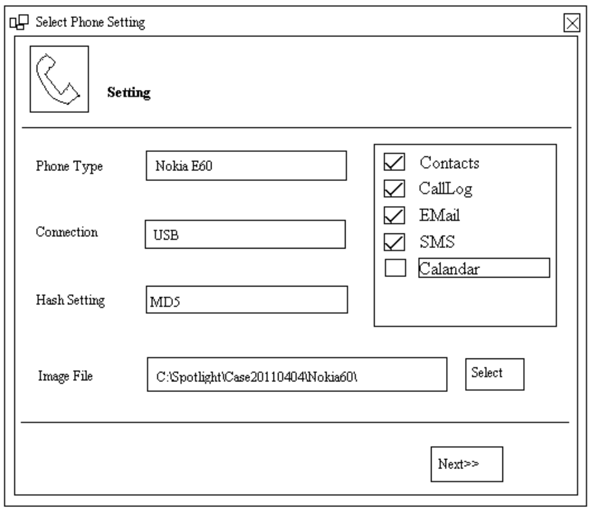
So far, if an application needs to have the complete access to the phone file system, it has to be authorized by means of the Symbian signing procedure with AllFiles capabilities, which requires a special certificate released by Trust Center. Three steps are required to sign an application. Initially, the installation file generator, make sis.exe, creates the installation files (extension.sis) from information specified in the package file (extension.pkg). After that, if the application is for the international market, it will be signed with an ACS Publisher ID, by means of the Symbian Signed service. Conversely, it will be signed with a user-generated certificate, which might be created with makekeys.exe. Finally, the Installation File Signer (signsis.exe), digitally signs the installation files with the proper certificate, by generating, as a result, a.sisx file.

5. PROPOSED DATA COLLECTION SCHEME FOR SYMBIAN SMARTPHONE

It suggests a distinct approach both in development and extracting of data from the device. Application is developed as a composable part – a part provides services to other parts and consumes services provided by other parts. Each plug-in exposes a contract identifier so that it can talk to other parts in the application. The data extracting is based on client server architecture. During acquisition, the tool should have full access to the object store, there are very severe constraints to obtain a full physical image, so a logical copy of the object store is proposed. The client part is installed on the desktop PC the server part is copied into the mobile device giving us API access to extract a copy. The first problem that had to be tackled is the deployment of the tool onto the device. A number of ways to place the agent based tool on the device were considered. The tool can be packaged as a SIS installer, so it can be sent to the phone using Bluetooth, infra-red or file transfer using the PC suite. A SIS file is a special software installer for the Symbian platform. Even though it may change certain parts of the file system, the changes are very little.

5.1 Data Acquisition

Data acquisition is the major step in forensics process. According to the forensics principle, the original data cannot be used for any forensics analysis. So we need to create a copy of the logical data present in the mobile device. This is achieved by developing an agent, which is to be installed on the target device. The module uses Symbian SDK, AT Command Set (SIM card commands), FBUS and Connectivity SDK to read the file system. The module is capable of establishing a connection and exchanging data with an externally connected host computer. Figure 5 shows setting screen of the acquisition GUI which allows the investigator to select the phone model, which further enables or disables the features available. Since the tool uses standard APIs to access the file system and uses an agent, which is the code signed by the signing authority, we can reasonably believe that these APIs will not change the device content. The main issue with this approach is that it requires a piece of software, called the agent, to be loaded on the device to acquire the content.



Phone Type: Nokia E60
Connection: USB
Hash Setting: MD5
Image File: C:\Spotlight\Case20110404\Nokia60\ Select

☑ Contacts
☑ CallLog
☑ EMail
☑ SMS
☐ Calandar

Fig5: Symbian Acquisition Process

5.1.1 Manual Examination

A manual examination is the simplest but most tedious means of capturing data from a smart phone device because of all the manual work involved. The examiner manually copies information through the device's native applications. This is performed if the investigator requires only a particular piece of evidence from the device [12] or when there is no other way to access that information. It is advisable that the examiner be familiar with the operation of the device in order to reduce room for error. This can be achieved by practicing on an identical test device or an emulator. It is important to record all actions taken on that device in order to allow the verification of the actions taken by others.

5.1.2 Connectivity Services

The most commonly used method of acquiring data from a mobile device is through the use of command-response protocols to interact with the Connectivity Services. Tools which employ this method either use open protocols such as the AT Command Set [20], SyncML [19] or OBEX [21], or proprietary protocols like the Nokia FBUS [7]. There exists wrappers or PC connectivity SDK's [13] which enable application developers to write programs which make use of mobile phone services without having to implement the underlying protocols. The most common tools which can access the device's memory are the manufacturer tools. These are software packages for data synchronisation of the device and computer. These applications are however not designed for forensics and in using them, there is a risk of altering the data on the phone if not used properly. Other non-forensic third party applications which can perform the same function also exist. The most widely accepted method for obtaining data from the mobile is the use of mobile forensic tools such as XRY, Oxygen Phone Manager - Forensic Edition and Paraben Device Seizure. These tools use the same protocols as the non-forensic tools but do not implement commands that explicitly modify the contents of the phone. This does not guarantee that their actions will not modify the contents of the phone [9] since a command, even though it is requesting data, may cause the operating system to change the item being requested.

5.1.3 Connection Agent

Another method of retrieving data from a mobile phone is the use of a connection agent. A connection agent is a small program that is placed on the target device to enable connection establishment and data exchange between the phone and the tool. This approach uses a client-server architecture with the agent acting as the server. Without the agent, the data retrieval tool is unable to retrieve the data from the device. This approach is similar to the one above, except instead of relying on the phone's connectivity services, custom connectivity services are used to gain access to the device's data. The one problem with this method is that it requires a piece of software to be loaded on the device, thus modifying the target device.

5.1.4 Direct Access

Directly accessing the phone's memory is the most forensically sound but also the most challenging [9] of the methods of retrieving data from a mobile phone. This method enables the imaging of the entire phone's memory, making it possible to access deleted or partially overwritten entries. It also bypasses the phone's security measures which would otherwise prevent access to the phone's contents. Another advantage to this method is that there is no reliance on the operating system to provide the correct results. This is not the case with the above mentioned approaches. Willasen [9] proposes two methods which are forensically sound:

1. Remove the phone's memory chip so it can be accessed directly.
2. Connect the phone to a motherboard to access the memory.

These proposed methods require a very high level of technical expertise which is usually not available to the average law enforcement agency. They require detailed knowledge of the phone's inner workings; therefore a fair amount of research is required in order to be able to perform proper analysis. It is therefore infeasible that these methods could be used by the average investigator since he will not possess the required knowledge.

5.1.5 Service Provider

An alternate means of getting information about a mobile phone is obtaining it from the service provider [17]. This is because a mobile phone typically functions in a GSM [14] network environment. When the network subscriber accesses a service from the network, it will be recorded by the service provider for billing purposes. If the examiner has the authority, he can request this information directly from the service provider. The information from the service provider will be more reliable than that from the mobile since it is more difficult to tamper with the service provider's records. It can also serve as a way to validate the data obtained from the mobile during an initial examination. The big drawback with this source of data is the limited amount of data that can be retrieved using this method. This is because there is a lot of data which will not be captured by the service provider either because of cost constraints or because it is not transmitted over the network.

5.2 Data Analysis

The tool creates a logical copy of the data present in the mobile device as a file at the desktop PC, where the client program is running. The tool also supports to generate the hash value, which will prove the authenticity of the acquired data. The image created can be loaded in an analyzer so that the data present in the mobile device can be viewed for further analysis. The tool provides important forensic information like Contacts, Call logs, SMS etc. This information will help the investigation agencies to get some cues for further investigation. The analysis tool shows all these information in separate file viewers. The incoming outgoing and missed call details are displayed separately. Also the Inbox, Outbox, Draft, Sent and Deleted SMS are categorized in separate viewers. The analysis tool is also provided with keyword and file search facility, which is the key feature of a forensics tool. User can add any keywords and file extensions in the box provided and the tool will search the entire image for the string entered. It shows the search hits in a separate viewer.

5.2.1 Other forensic tools which support Symbian OS based devices

Followings are the other forensic tools which support Symbian OS based devices,

- XRY:

XRY is a forensic software toolkit that supports a variety of mobile phone devices. It uses SyncML and OBEX for Series60 based devices and a connection agent for UIQ based Symbian devices. For the purposes of this paper, XRY version 3 was tested and it is able to retrieve the following data when using SyncML and OBEX:

1. Contacts
2. Notes
3. Videos
4. Audio
5. Pictures

SyncML was used to retrieve all the above items except for the file system. OBEX was used to retrieve the files and only a very small subset of the file system was retrieved. When the connection agent was used with a UIQ based device, most of the files were retrieved. Some files however could not be retrieved and this is probably because of operating system restrictions. All the files which could not be retrieved using the on-phone tool, also could not be retrieved using the XRY connection agent. The connection agent, like the on-phone tool, could not retrieve deleted information.

- Oxygen Phone Manager:

The original Oxygen Phone Manager (OPM) is a cell phone management tool; the difference between the standard and the forensic version is that the forensic version prohibits modifications to the target device. OPM uses a connection agent [5] to enable interaction with the device. There are six versions of the agent, one for each flavor of the operating system. OPM allows the extraction of the following data:

1. Call information
2. Messages
3. Connection logs (Wi - Fi and GPRS)
4. PIM related data
5. Installed applications and games information
6. Multimedia
7. Flash card files

When evaluated against the requirements for a mobile phone examination tool, OPM and XRY do not fare very well in fulfilling the requirements. They both require an agent to be placed on the device, which violates the first requirement; and they also require a lot of interaction with the device to get the agent to connect with the workstation. This is especially true for OPM, which requires the most amount of interaction to successfully establish a connection. There is one drawback to the on-phone tool that the other tools also suffer from. Even though the tool does not explicitly initiate writes to the device, it does not prevent the operating system or other system processes from modifying the device's content. For example, if the device is not set to flight mode, it can still receive new messages. This, in most cases is not desired since the state of the device must be preserved by as much as possible after taking it into custody.

6. ADVANTAGES & DISADVANTAGES

6.1 Advantages of Symbian OS

- Provision of security:
  They support many types of security-applications like anti-theft and anti-virus applications.
- Elegant design:
  It has the elegant design, created from the start for embedded systems, with limited resources and therefore with fine grained control over them.
- Office compatibility:
  The Symbian operating system incorporates full support for Word, Excel, and PowerPoint documents, though again the ability to create and edit these documents or just view them depends on hardware.
- E-mail:
  Symbian is as versatile as any other platform. It supports Webmail accounts. The operating system also supports Lotus Notes and Microsoft Exchange for maximum compatibility in the corporate world.
- Multimedia:
  The Symbian operating system is pretty adept at multimedia, with integrated support for audio and video playback and recording. The Nokia 9300, for instance, comes with both an MP3 player and the mobile version of RealPlayer, which enables playback of RealAudio, Real Video, and MP3 files.
- Third-party applications:
  A recent check of software site Handango revealed more than 5,500 third-party applications for the Symbian operating system--not quite up to Palm and Windows Mobile levels, but far more than you'll find on other smart phone operating systems. That's another big point in Symbian's favor, especially if you're weighing it against other phones.

6.2 Disadvantages of Symbian OS

The Symbian OS covers more features than any other OS, so feature-wise there's not much missing. However there are plenty of issues with the Symbian platform and most of them will fall on the shoulders of developers.

- Steep learning curve for developers :
  Developers find it difficult to work with Symbian because of poor documentation, development tools and also because of a massive library with thousands of classes. Also, Symbian programming paradigms differ from conventional methods so that programmers may find it difficult to understand and get the hang of it.
- Porting:
  Porting between devices can be tricky. There is a need to take the OS version and device-specific restrictions into account. For instance, one cannot port an application that is built for the UIQ Platform directly to an S60 UI Platform because there is a need to take the touch-screen dependencies into account. There might even need to take some issues into consideration when porting between applications built for the same platforms but for different devices.

6.3 Advantages of Forensic Process

- Recovery of deleted text:
  It is useful to recover deleted text messages, photographs, videos, address books, caller ID's, contact information, and several other forms of data, that can uncover employee theft, bullying, infidelity, and the use of illegal drug substances.
- Easy identification:
  Because of the development of mobile phone forensics, law enforcement can more readily identify pedophiles, stalkers or harassers through mobile phone forensics.
- Can find Geographical location easily:
  Using Phone Forensic Process Geographical location of the phone at any given time including history of where the phone has been.

6.4 Disadvantages of Forensic Process

- Possibility of tampering:
  If the phone is left on it is possible to tamper with the evidence causes problem in forensic process.
- Difficult to identify exact phone:
  There are many different manufactures which all offer a host of different models with new models becoming available almost monthly, so identifying the exact phone is difficult.

7. APPLICATIONS OF SYMBIAN OPERATING SYSTEM

- Y-Browser:
  Y-Browser is a free file manager application designed for S60 3rd edition Symbian smart phones. It implements most standard features on files (such as copy, cut, paste, etc) & folders (create, remove, etc) and it allows to work with "hidden, system" folders.
- AutoLock:
  AutoLock is a free automatic key lock application. It will turn key lock on after certain amount of time of inactivity. Its features are,
  1. Automatic start-up when the phone is turned on.
  2. Setup tool to change inactivity time.
  3. It's running completely on the background.
  4. It's free.
- QuickMark:
  It is a 2-D barcode system designed for mobility electronic devices with camera module (such as camera phones, smart phones, Webcams, scanners, etc.). QuickMark can simplify the original tedious operate process on cell phones.

8. CONCLUSION

The fast development in Smartphone device technology is making the digital forensic of these devices a very complicated task. Also, this development leads to increasing problems in developing and maintaining a scientifically sound method for the capturing of data from these devices. The methods commonly used to acquire data from mobile phones are software based implementations of communication protocols such as AT commands, OBEX, SyncML and Nokia FBUS. Hence an agent based approach is useful for reliable and faster extraction of data from the mobile phones so that it is accepted by the court.

9. REFERENCES

[1] Michael Aubert, with Alexey Gusev ... [et al.], Quick Recipes on Symbian OS, Mastering C++ Smartphone Development. John Wiley & Sons, Ltd, 2008. pp. 529–551.

[2] Symbian-Ltd. Carbide. C++: Introductory White Paper. Forum Nokia, Version 1.1; 2007 13

[3] Breeuwsma M., Jongh M. D., Klaver C., et al. Forensics data recovery from flash memory. In Small Scale Device Forensics Journal, 2007, 1. 7

[4] Iain Campbell, Symbian OS Communications, John Wiley and Sons, Ltd. , 2007. 11

[5] [Software, 2007] Software, O. (2007). *Oxygen Phone Manager II Forensic Edition v2.13 for Symbian OS smartphones*. Oxygen Software, http://www.oxygensoftware.com/en/, 2.13 editions.

[6] Craig Heath, *Symbian OS Platform Security: Software Development Using the Symbian OS Security Architecture* (Wiley, 2006).8

[7] [Kot and Zoltan, 2006] Kot, P. and Zoltan, B. (2006). Gnokii project.

[8] [Siezen, 2005] Siezen, S. (2005). Symbian OS version 9.1 product description. Technical Report 1.1, Symbian Ltd.

[9] [McCarthy, 2005] McCarthy, P. (2005). Forensic analysis of mobile phones. Master's thesis, University of South Australia.

[10] [Willassen, 2005] Willassen, S. (2005). Forensic analysis of mobile phone internal memory. In *IFIP Int. Conf. Digital Forensics*, pages 191–204.

[11] Jo Stichbury Symbian OS Explained, John Wiley and Sons, Ltd. 2004, 1. 10

[12] [Casey, 2004] Casey, E. (2004). *Digital Evidence and Computer Crime*. Elsevier Academic Press, 2nd edition.

[13] [McDowall, 2004] McDowall, I. (2004). *Programming PC Connectivity Applications for Symbian OS*. Wiley.

[14] [Croft, 2004] Croft, N. (2004). Secure interoperation of wireless technologies. Master's thesis, University of Pretoria.

[15] [Mery, 2003] Mery, D. (2003). Symbian os version 7.0 functional description. Technical Report 1.5, Symbian Ltd.

[16] Svein Yngvar Willassen, Forensics and the GSM mobile phone system, The International Journal of Digital Evidence, Spring 2003, Volume 2 Issue 1

[17] [Goode, 2003] Goode, A. (2003). Forensic extraction of electronic evidence from gsm mobile phones. *Secure GSM and Beyond*, 10059(9):1–6.

[18] OMA (2001).Syncml sync protocol. Technical Report 1.0.1, Open Mobile Alliance

[19] [OMA, 2001] OMA (2001). Syncml sync protocol. Technical Report 1.0.1, Open Mobile Alliance.

[20] [SMG, 1999] SMG (1999). Digital cellular telecommunications system (phase 2) - at command set for GSM mobile equipment (me). Technical Report ETS 300 642, European Telecommunications Standards Institute, 650 Route des Lucioles - Sophia Antipolis - Valbonne – FRANCE

[21] [McGowan et al., 1999] McGowan, P., Suvak, D., and Kogan, D. (1999). IrDA object exchange protocol. Technical report, Infrared Data Association.

[22] Richard Harrison, Mark Shackman, Symbian OS C++ for Mobile Phones Volume 3. John Wiley and Sons, Ltd. pp. 204-206

[23] Michael Auburt, Quick Recipies on Symbian OS Mastering C++ Smartphone Development, John Wiley and Sons, Ltd. pp. 60-63