AC 2008-2195: THE DEVELOPMENT OF A FORENSICS TOOL FOR WINDOWS MOBILE DEVICES

Kyle Lutes, Purdue University

Kyle Lutes is an Associate Professor of Computer & Information Technology (CIT). He has authored/co-authored numerous papers, many of which were presented at national conferences or published in trade magazines/journals as well as two college textbooks. His background and interests cover all areas of software development, including mobile computing, client/server information systems, web application development, object-oriented programming (OOP), programming languages, software engineering, user interface design, and rapid application development (RAD). Kyle has been writing software professionally since 1982. Prior to his current appointment at Purdue, he held various software development positions in industry and has worked on projects for such industries as banking, telecommunications, publishing, hospitals, medical schools, retail, and pharmaceuticals.

Addition to his teaching and research duties at Purdue, Kyle is the founder of DelMar Information Technologies, LLC. His company specializes in custom software development using Microsoft technologies (C#, .NET, .NET Compact Framework, Active Server Pages (ASP), SQL Server, and Visual Basic) for mobile devices (smart phones and Pocket PCs), enterprise, web, client/server and desktop architectures. DelMar Information Technologies also sells several software products and services, including corporate training classes.

Richard Mislan, Purdue University, West Lafayette

Richard is an assistant professor specializing in the area of Cyber Forensics. Richard 's areas of research include Small-Scale Digital Device Forensics, Unusual Sources of Digital Evidence, and the Application of Artificial Intelligence Techniques for Improving Efficiency in Cyber Forensics. He is also a faculty member with the Center for Education and Research in Information Assurance and Security (CERIAS).

Formerly of Ferris State University, Big Rapids, Michigan, Richard has taught graduate and undergraduate courses in Information Security, Network Management, and E-Business Strategy. Richard has also served as a Technology Director and Educator for various school districts, a Communications Electronic Warfare Officer for the U.S. Army, and a Radio Disc Jockey.

He has authored several articles in the area of Small Scale Digital Device Forensics, serves as co-editor for the Small Scale Digital Device Forensics Journal, and acts as a reviewing editor for the National Institute of Standards and Technology (NIST) on Guidelines for Cellphone Forensics, Guidelines for PDA Forensics, Cell Phone Forensic Tools, and PDA Forensic Tools. Rick is in the process of completing his doctoral dissertation, preparing the ultimate online resource for his course in Small Scale Digital Device Forensics, and planning the annual Mobile Forensics World Conference.

WinMoFo: The Development of a Forensics Tool for Windows Mobile Devices

Abstract

The ubiquity of mobile computing devices (e.g. smartphones), our society's ever increasing use of these devices, and the continual appearance of these devices at crimes scenes has created a need for tools to aid in the acquisition of critical, time-sensitive evidence. The term "mobile forensics" is used to describe the acquisition and analysis of data found on mobile computing devices, as well as the data on the SIM/USIM cards and other memory cards they contain. The retrieved data can then be used in the aid of an investigation or in a court of law. Multiple documented procedures are in place and must be adhered to in the forensics acquisition and analysis of mobile phone data. One of the largest issues surrounding mobile phone forensics is the proprietary methods of storage used by each phone manufacturer.

Many different mobile devices are based on the Windows Mobile operating system from Microsoft. In addition to basic voice capabilities, Windows Mobile devices can be used to store contacts, calendar appointments, emails, text messages, and call histories. Additionally, because these devices frequently include a digital camera, they can store digital photos and video files. Currently, there is just one software tool designed to help law enforcement officers with the acquisition of information contained on Windows Mobile devices. However, this tool is part of a larger forensic software package and its price puts it out of the reach of many potential users.

In this paper we first provide an overview of the trials and tribulations associated with mobile forensics. Secondly, we describe our reasoning for developing our proof of concept software tool which can be used to acquire nearly all data from Windows Mobile devices. Data retrieved from the device can be displayed on a connected laptop computer, saved for later analysis, or printed. Third, we list the technologies used for its development. Finally, we conclude with a demonstration of the software and our future plans for its continued development.

The Ubiquity of Mobile Computing Devices

Following in the steps of PDAs, smartphones are becoming personal oracles of information [1, 2]. While early generation cellular telephones were used only for voice communications, modern digital mobile phones have quickly become societal necessities for daily existence. Not only do smartphones support voice communications, these devices provide technologies for Short Message Service (SMS) messaging, Multi-Media Messaging Service (MMS) messaging, Instant Messaging (IM), electronic mail, Web browsing, multimedia capturing and playback, electronic document previewing, basic Personal Information Management (PIM) applications (e.g., contacts, calendar, etc.) and financial transactions.

The use of smartphones by consumers continues to grow. Consider these recent data points:

- For the July to September 2007 quarter, market research group NPD reported US sales of 4.2 million smartphones, a 180% increase over the same quarter last year[3].

- A recent Bloomberg report shows the sales of smartphones almost tripled last quarter and made up 11 percent of all phones sold in the U.S. Shoppers spent $3.2 billion on phones, or $83 each, up from $2.2 billion a year earlier[4].
- Apple claims to have sold four million iPhone smartphones during 2007 which is about half of their goal of selling 10 million iPhones by the end of 2008[5].

Additional entries into the smartphone space by Google with their Android smartphone platform, and Yahoo! with their *Yahoo! Go 3.0 beta* are additional key indicators that the industry heavyweights are expecting huge increases in the number of smartphone users.

The Need for Forensic Tools

As society gravitates towards the adoption of such technologies, so does the criminal population. Mobile computing devices have been found at numerous crimes scenes around the world, usually as corroborative evidence or investigative leads. In the world of digital forensics, law enforcement investigators are just now realizing the potential of the evidence that can be gleaned from smartphones[1]. Such evidence includes contacts, calendar appointments, emails, text messages, call histories, digital photos, and videos.

Smartphones are used by drug dealers to manage contacts, child pornographers to store digital photos, and sexual predators as an instant messaging device. Furthermore, digital photos found on smartphones have helped convict suspected criminals. For example, murderers and rapists have used smartphones to take so-called "trophy shot" digital photos of their victims, and youths have recorded videos of themselves committing acts of sexual assault and vandalism.

Digital forensic examiners need a toolkit that specifically acquires and accurately presents digital evidence from mobile computing devices. Primarily there are two types of digital evidence collection that are instrumental in any investigation, "On Scene" and "In the Lab." On scene, from any mobile phone, it is imperative to collect contacts, call history, and text messages. These sources of personal information give contextual clues to the next steps of any investigation; it identifies who you know and to whom you talk. The other set of digital evidence comes from a more lengthy process conducted later in a forensic lab. This information comes from the images, videos, web browser cache, and other document files found on the device. Finally, the means to report and export the analyzed evidence collected is important for the investigation[6].

Unfortunately, the currently available digital forensic examiners toolkits are light on tools to aid in the acquisition of information from smartphones. To advance the mobile device forensics field out of its current infant stage, numerous challenges must be addressed by the makers of forensics tools. These challenges can be classified into six general categories: 1) the many manufacturers of mobile phones, 2) preserving data on the device, 3) the nuisance created by the need to carry to every investigation the many varieties of power and data connectors, 4) the various operating systems and communication protocols used by the device vendors, 5) security mechanisms on the mobile device, and 6) the unique data formats used to store information on the device. Finally, to add to this frustration, some phones, notably pre-paid or "Pay As You Go" phones, do not provide the means for data connectivity.

Even though the number of smartphone device models and manufacturers is very large, they all are based on only a handful of operating system (OS) options. In the third quarter of 2007, more than 75% of all smartphones sold in North America were either RIM's Blackberry, Apple's iPhone, or smartphones based on Microsoft's Windows Mobile OS. (Note, because the Windows Mobile OS is based on the Microsoft Windows CE general-purpose OS, some statistics group market share by Windows CE devices rather than Windows Mobile devices.) The Blackberry still leads the pack, but the iPhone has quickly gained 27% of the market in North America to claim the number two spot, while Windows Mobile devices remain a respectful third with approximately 24% of the market[7].

Given the sizable market held by Windows Mobile smartphones and the need for forensics tools by law enforcement agencies, one might imagine many such forensics tools would be available. However, only one tool is currently available that includes software specifically made to acquire information from Windows Mobile devices. Paraben Corporation offers a commercial product named Device Seizure that can acquire and analyze information from many mobile phones and other handheld computing devices, including those running the Windows CE and Windows Mobile OSs. However, because the Paraben tool is made to work with many different types of mobile devices, including non-Windows devices, it suffers from not always forensically acquiring and analyzing any one device completely[8]. Additionally, the Paraben tool's high price of ~$900 puts it out of the reach of many potential users.

WinMoFo Conceptualization

We authors both work in the Department of Computer and Information Technology at Purdue University. Kyle Lutes joined the faculty in 1998, teaches software development courses, has over 25 years experience in the software development field, and has most recently been specializing in application software development for mobile computing devices. Rick Mislan joined the faculty in 2006 and is well known as a national expert in the emerging field of small scale digital device forensics. Both authors share a common interest in mobile computing device and have collaborated on several mobile computing related projects.

During one of these projects, Professor Mislan discussed the dearth of tools available to help investigators acquire information from smartphone devices. After a few discussions of desired features, and a few "bar napkin design sessions", Professor Lutes performed a several short experiments to determine the feasibility of the authors developing such a tool themselves. When the experiments proved promising, we decided to proceed with developing a proof of concept forensics tool for Windows Mobile devices.

Within four days of heavy coding and testing, Kyle had just over 2,000 lines of hand-typed source code and had created WinMoFo – the working proof of concept Windows Mobile forensics application. The relatively short development time and small number of lines of source code can be attributed to the software development tools used.

Technologies Used to Develop WinMoFo

Microsoft has a long history of making tools available to software developers that enable them to relatively easily create applications for Microsoft operating systems. The mobile computing space is no different. Shortly after Microsoft released their version of the new programming language C# and related .NET Framework class library and run-time environment, they announced the .NET Compact Framework (.NET CF). The .NET CF is a much smaller version of the full .NET Framework and was designed to aid in the development of software for small, resource constrained devices. Because many of the .NET CF APIs are the same as the full .NET Framework, sharing of source code between desktop and mobile applications is possible[9].

Professor Lutes began using C# with the .NET CF during the summer of 2002 when it was still in beta. He has continued to use both C# and the .NET CF to write Window Mobile applications and had enough knowledge about each to believe they could be used to develop a Windows Mobile forensics tool in a relatively short amount of time.

One drawback of using the .NET CF class library is that it was designed to be a scaled-down version of the full .NET Framework. Because it was not designed specifically for mobile application development, it does not contain APIs for any of the smartphone specific features. For example, no .NET CF APIs exists to access a phone's call log or text message data store. However, because C# does allow direct OS calls via a process known as *P/Invoke*, we reasoned our C# code should be able to access all the information on the smartphone we desired. This proved much more difficult than expected.

Despite Windows CE being available since 1996 and the .NET CF since 2002, extensive documentation on the Microsoft MSDN web site, and several books published on the subject, non-trivial code examples for accessing smartphone features are still difficult to find. After a day of struggling with writing code to access a smartphone's call log, we found an excellent 3rd party library that had all the features we needed.

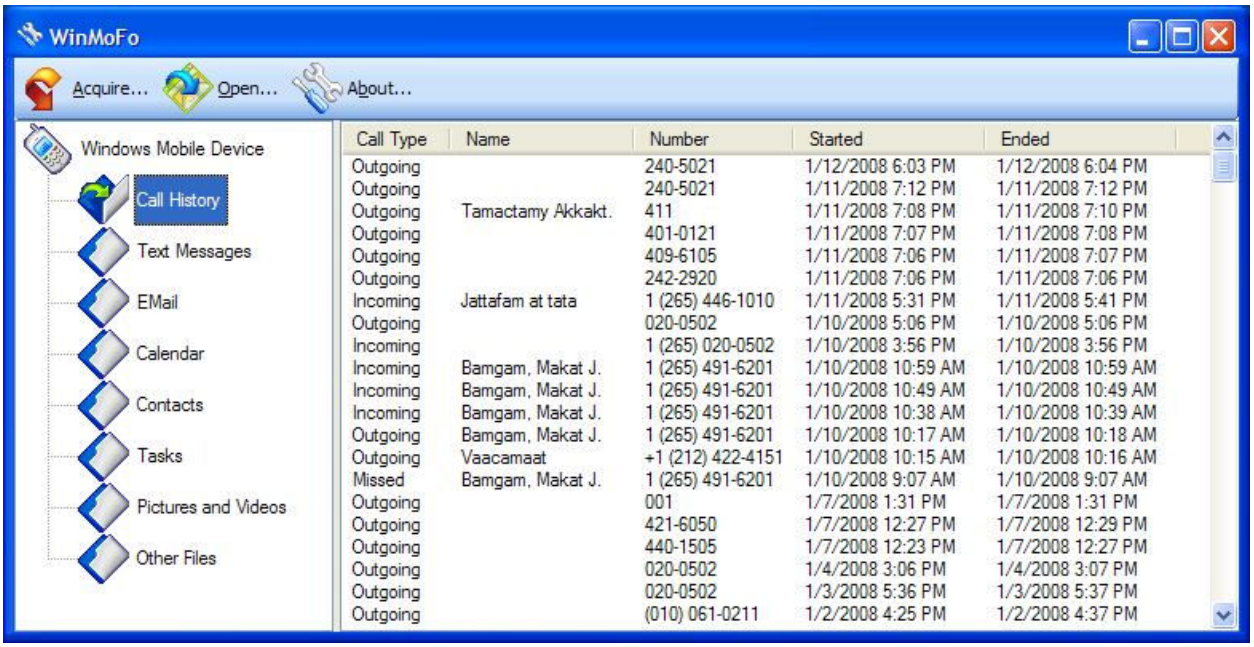
The "Mobile In The Hand 3.0" product from In The Hand Ltd. proved essential to us in finishing our proof of concept forensics tool in the time we allotted for development. Mobile In The Hand is distributed as a dynamic linked library (InTheHand.DLL) that contains classes and methods to easily allow C# source code to access smartphone resources[10]. In short, it contains the smartphone APIs that the .NET CF lacks.

One other Microsoft technology we used is known in the software development world as RAPI (pronounced "rappy"), which is short for Remote API. The RAPI DLL allows desktop PC applications to manipulate the file system on smartphones that are attached to the PC. For example, RAPI methods exist to copy files to and from the device, execute programs on the device, and access the smartphone's registry entries[11].

Using WinMoFo

To use WinMoFo, one needs a Windows (2000, XP, or Vista) laptop or desktop PC, the WinMoFo executable, a smartphone based on Windows Mobile, and a USB data cable to

connect the smartphone to the PC. The system requirements for WinMoFo are very modest. For versions of Windows previous to Vista, Microsoft ActiveSync (a free download) must be installed to enable communication between the PC and phone. (Note, Windows Vista ships with an ActiveSync replacement named Windows Mobile Device center.) The only remaining requirement is that the Microsoft .NET Framework must be installed on the Windows PC.



Call Type	Name	Number	Started	Ended
Outgoing		240-5021	1/12/2008 6:03 PM	1/12/2008 6:04 PM
Outgoing		240-5021	1/11/2008 7:12 PM	1/11/2008 7:12 PM
Outgoing	Tamactamy Akkakt.	411	1/11/2008 7:08 PM	1/11/2008 7:10 PM
Outgoing		401-0121	1/11/2008 7:07 PM	1/11/2008 7:08 PM
Outgoing		409-6105	1/11/2008 7:06 PM	1/11/2008 7:07 PM
Outgoing		242-2920	1/11/2008 7:06 PM	1/11/2008 7:06 PM
Incoming	Jattafam at tata	1 (265) 446-1010	1/11/2008 5:31 PM	1/11/2008 5:41 PM
Outgoing		020-0502	1/10/2008 5:06 PM	1/10/2008 5:06 PM
Incoming		1 (265) 020-0502	1/10/2008 3:56 PM	1/10/2008 3:56 PM
Incoming	Bamgam, Makat J.	1 (265) 491-6201	1/10/2008 10:59 AM	1/10/2008 10:59 AM
Incoming	Bamgam, Makat J.	1 (265) 491-6201	1/10/2008 10:49 AM	1/10/2008 10:49 AM
Incoming	Bamgam, Makat J.	1 (265) 491-6201	1/10/2008 10:38 AM	1/10/2008 10:39 AM
Outgoing	Bamgam, Makat J.	1 (265) 491-6201	1/10/2008 10:17 AM	1/10/2008 10:18 AM
Outgoing	Vaacamaat	+1 (212) 422-4151	1/10/2008 10:15 AM	1/10/2008 10:16 AM
Missed	Bamgam, Makat J.	1 (265) 491-6201	1/10/2008 9:07 AM	1/10/2008 9:07 AM
Outgoing		001	1/7/2008 1:31 PM	1/7/2008 1:31 PM
Outgoing		421-6050	1/7/2008 12:27 PM	1/7/2008 12:29 PM
Outgoing		440-1505	1/7/2008 12:23 PM	1/7/2008 12:27 PM
Outgoing		020-0502	1/4/2008 3:06 PM	1/4/2008 3:07 PM
Outgoing		020-0502	1/3/2008 5:36 PM	1/3/2008 5:37 PM
Outgoing		(010) 061-0211	1/2/2008 4:25 PM	1/2/2008 4:37 PM

Figure 1 – The WinMoFo desktop PC user interface showing sample call history acquired from a Windows Mobile (personal information obfuscated)

After starting WinMoFo, the user simply connects the smartphone to the PC using a USB data cable and clicks the Acquire toolbar button. Next, the user is prompted to select a destination folder on the PC in which to store the device files. After the destination folder is selected, the user waits a short time for the data on the smartphone to be copied to the PC. Acquisition times vary depending on how much data is stored in the device, but our tests have shown most acquisitions take just one or two minutes.

Our current version of WinMoFo can retrieve detailed information on voice call logs, SMS text messages, email messages, calendar appointments, contact information, the task list, and a list of all files found in the device's file system. Data is displayed to the user using a familiar folder hierarchy as can be seen in Figure 1 and Figure 2. Once acquisition has finished, the user can quickly switch between the various types of information gathered, and scroll and sort the information in the lists. Because WinMoFo stores data in simple ASCII text files on the PC, the data can be viewed with other applications (e.g. spreadsheets and databases), copied to external storage (e.g. USB flash memory devices), and printed.

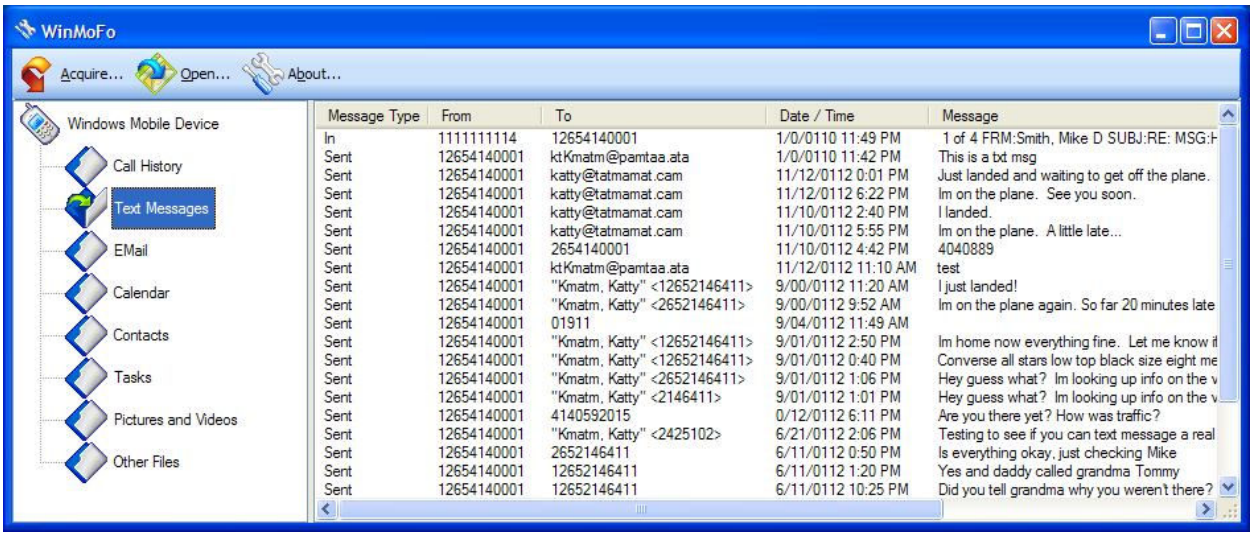


Figure 2 – The WinMoFo desktop PC user interface showing sample SMS text message history acquired from a Windows Mobile smartphone (personal information obfuscated)

Most smartphone devices allow additional memory to be added in the form of mini and micro secure digital (SD) cards. These tiny cards can easily store gigabytes of information at a very low cost. Two gigabyte cards now cost less than $20.00. Because most smartphones have the capacity to store large amounts of data and likely will be populated with large media files, WinMoFo does not automatically copy all files from the device to the PC. Rather it first creates a list of all files found on the device, including those stored on any expansion memory cards, and displays the file names to the user. The user can then choose to copy any or all files to the PC.

As was previously stated, digital photos and videos are sometimes used by criminals to document their crimes. To make this type of potential evidence easy to find by investigators, WinMoFo displays these types of files separate from the other non-photo and video files. From either list of files, the user can double-click a file to copy it to the PC, view its contents, or select many files and copy them all to the PC by using the right-click context menu as shown in Figure 3.

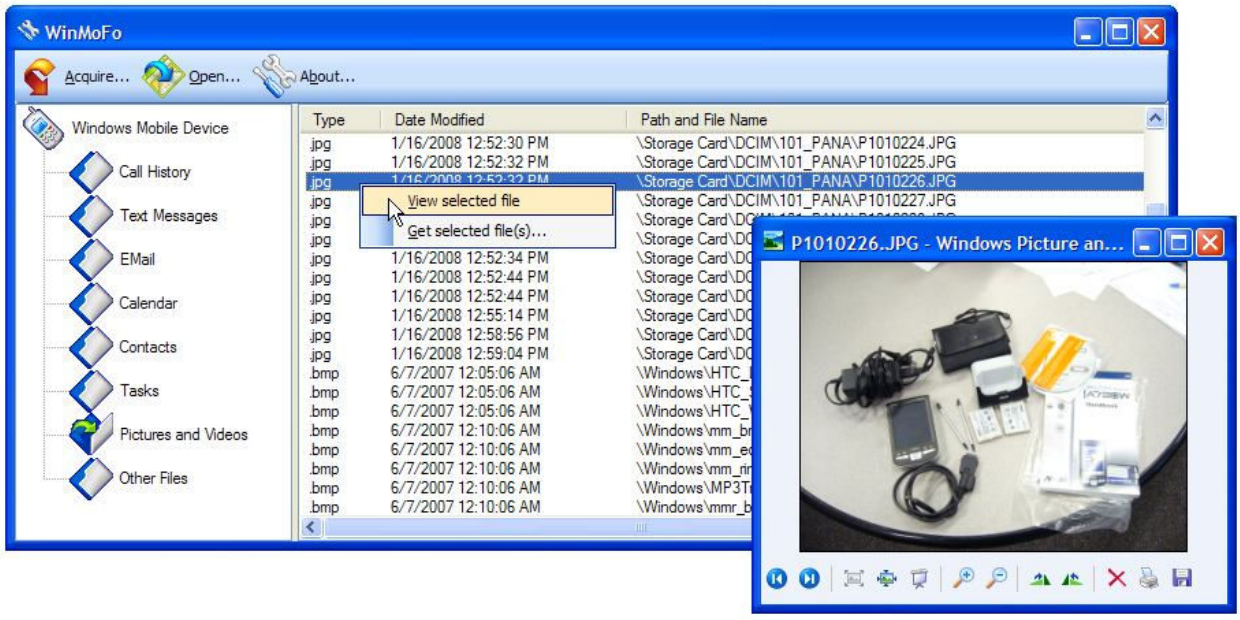


Figure 3 – The WinMoFo desktop PC user interface showing an image file being viewed from an acquired Windows Mobile smartphone

How WinMoFo Works

WinMoFo consists of two separate programs. The main desktop PC user interface is handled by WinMoFoPC. A much smaller helper program, WinMoFoCE, runs on the mobile device to gather information from the various sources and send it back to WinMoFoPC.

When the user clicks Acquire on the toolbar button, WinMoFoPC uses the RAPI API to verify connectivity to the attached smartphone, copies the WinMoFoCE files to the device, and starts it up. While WinMoFoCE is starting, WinMoFoPC opens a TCP socket and waits for a connection. When WinMoFoCE starts, it first opens a TCP socket connection back to WinMoFoPC which it uses to send the data. Communication via a TCP socket connection between the smartphone and PC is possible because when a Windows Mobile device is connected to the PC, Activesync will assign it an IP address.

WinMoFoCE then enumerates through the various data stores for call logs, SMS text messages, email, calendar, contacts, and so on. The InTheHand DLL makes this enumeration process a trivial *for* loop:

```
for (int i = 0; i < callHistory.Count; i++)
{
 CallDetails call = (CallDetails)callHistory[i];

 stream.Write(call.CallType + "\t"
   + call.Name + "\t"
   + call.Number + "\t"
   + call.StartTime.ToShortDateString() + " "
   + call.StartTime.ToShortTimeString() + "\t"
   + call.EndTime.ToShortDateString() + " "
   + call.EndTime.ToShortTimeString());
}
```

File system information is gathered without the aid of the InTheHand DLL as the .NET CF has a very robust API for accessing and manipulating the files and folders on the device. For example, getting a list of all files in a folder can be done using a simple *foreach* loop. To get all files in all folders on the device, a simple recursive iteration is used:

```
private static void getWinMoFoFiles(string rootFolder)
{
 foreach (string file in Directory.GetFiles(rootFolder))
 {
   mStream.Write(file + "\t" + File.GetLastWriteTime(file));
 }
 ArrayList folders = new ArrayList();
 foreach (string folder in Directory.GetDirectories(rootFolder))
 {
   folders.Add(folder);
 }
 foreach (string folder in folders)
 {
   getWinMoFoFiles(folder);
 }
}
```

As the information is gathered on the device, WinMoFoCE sends it to the PC using the open socket connection. WinMoFoPC continually updates the UI as information is received from WinMoFoCE. This process continues until all necessary information has been sent, at which time WinMoFoCE sends WinMoFoPC a message indicating it has finished. When WinMoFoPC receives notification that WinMoFoCE has ended, it uses the RAPI API one final time to remove the WinMoFoCE files from the device.

WinMoFo Limitations

While our original intentions were to develop WinMoFo as a proof of concept application, we feel we have advanced it past the proof of concept stage. However, we do not feel confident that it is yet robust enough to release as a public beta.

Most importantly, WinMoFo can benefit from more testing. We have tested WinMoFo using several of the Windows Mobile 5 and Windows Mobile 6 smartphones we use and have in our labs, but testing with a more varied set of phones from real users will help uncover remaining bugs, as well as bring to our attention any problematic differences between the phone models from various hardware manufacturers and carriers. Likewise, having real law enforcement officers test the user interface will help identify any confusing or missing features.

As with many software products, graceful handling of exceptions needs to be improved. For example, WinMoFo works very well when all logic executes as planned, but when an unexpected event occurs, such as the data cable becoming disconnected during processing, WinMoFoPC must be closed and WinMoFoCE manually removed from the smartphone.

Because we developed WinMoFoCE using C# and .NET, the smartphone must have the .NET CF installed. This so far has not caused us any problems since normally some version of this framework is bundled with the Windows Mobile OS. Indeed, the choice to use C# and the .NET CF allowed us to complete our work so far in a very short amount time. However, we feel

WinMoFoCE might benefit from being rewritten using C++. A C++ version would eliminate any concerns about which version of the .NET CF is on the device, remove the dependency on the 3rd party InTheHand DLL, and likely reduce execution time.

A final task we hope to complete in the near future is to take steps to make WinMoFo easily available for potential users. At a minimum WinMoFo needs a complete setup package installer, proper certificate-based authentication, and a web site with a discussion forum to allow us to get feedback on problems users encounter and new features we should add.

Future Work

As previously mentioned, we intend to continue testing WinMoFo using additional devices and improving the code to make it more robust. The computer forensics group in our department frequently holds workshops and training sessions for law enforcement officers. Our department also has contacts at the National White Collar Crime Center and with the Chicago Police Department. We plan to demonstrate WinMoFo at these workshops and to our law enforcement contacts and solicit feedback. Additionally, we will be presenting WinMoFo at the Mobile Forensics World 2008 conference[12] held in Chicago, IL during the month of May.

Assuming no unforeseen roadblocks arise, we hope to make a public beta of WinMoFo available during the fall of 2008. If WinMoFo proves to be a popular tool to aid in mobile device forensics, we hope to secure funding that will allow us to adapt our technologies to work with the other two smartphone market leaders – the Apple iPhone and RIM's Blackberry.

Conclusions

Research statistics show continued explosive growth in the use of smartphone devices by the general public. As smartphone use increases among the general population, so does smartphone use increase by criminals. In addition to basic voice capabilities, smartphones can be used to store contacts, calendar appointments, emails, text messages, and call histories. Additionally, because these devices frequently include a digital camera, they can store digital photos and video files. Law enforcement officers are now just beginning to learn how to glean information and possible evidence from these mobile computing devices.

As the field of mobile device forensics matures, software tools are being developed to aid in acquiring data from smartphones and other mobile devices. However, even though smartphones based on the Windows Mobile OS accounted for about 25% of all smartphone sales in North America during 2007, only one forensics software tool designed specifically for Windows Mobile devices is currently available.

In this paper, we have shown that by combining off the shelf software development tools along with a good working knowledge of Windows Mobile application software development, we were able to develop a proof of concept software tool that can acquire nearly all information from a Windows Mobile smartphone. Our tool, named WinMoFo, is currently a work in progress but with further testing, feedback from potential users, and a few enhancements, we should be able to make WinMoFo beta available to the public.

Bibliography

1. Schenke, J. (2006). Portable digital devices create opportunities for cops and robbers. Retrieved December 8, 2007 from http://www.purdue.edu/UNS/html3month/2006/060316.Mislan.digital.html
2. Shachtman, N. (May 3, 2006). Fighting Crime With Cellphones' Clues. The New York Times. Retrieved December 12, 2007 from http://query.nytimes.com/gst/fullpage.html?res=9B00EED8113FF930A35756C0A9609C8B63&sec=&spon=&pagewanted=all
3. iPhone Grabs 27% of US Smartphone Market, RoughlyDrafted Magazine. Retrieved January 17, 2008 from http://www.roughlydrafted.com/2007/11/21/iphone-grabs-27-of-us-smartphone-market/
4. Heiskanen, V. (December 20, 2007). Mobile-Phone Spending in U.S. Sets Record on IPhone. Bloomberg.com. Retrieved January 17, 2008 from http://www.bloomberg.com/apps/news?pid=20601109&sid=aAiaFs7Y6jXo&refer=home
5. Block, R. (January 15, 2008). Live from Macworld 2008: Steve Jobs keynote. Retrieved January 17, 2008 from http://www.engadget.com/2008/01/15/live-from-macworld-2008-steve-jobs-keynote/
6. Jansen, W., Ayers,R. (2007). Guidelines on Cell Phone Forensics. Retrieved September 10, 2007 from http://csrc.nist.gov/publications/nistpubs/800-101/SP800-101.pdf
7. Padilla, A. (December 18, 2007). iPhone rapidly gains market share in North America. Retrieved January 17, 2008 from http://www.wirelessinfo.com/content/iPhone-rapidly-gains-market-share-in-North-America.htm
8. Paraben (2007). Device Seizure v1.3. Retrieved December 15, 2007 from http://www.paraben-forensics.com/catalog/product_info.php?cPath=25&products_id=405
9. Lutes, K., Change, K. (July 2007). Cross Platform Development for PDAs and Tablet PCs. SmartPhone & Pocket PC magazine. Retrieved January 17, 2008 from http://www.pocketpcmag.com/_archives/Jun07/crossdev.aspx
10. In The Hand Ltd. Mobile In The Hand version 3.0. Retrieved January 17, 2008 from http://inthehand.com/content/Mobile.aspx
11. Microsoft (2008). Remote API (RAPI). Retrieved January 17, 2008 from http://msdn2.microsoft.com/en-us/library/aa920177.aspx
12. Mobile Forensics World 08. Retrieved January 17, 2008 from http://mobileforensicsworld.com/