Forensic Extractions of Data from the Nokia N900

Mark Lohrum

Purdue University Cyber Forensics
West Lafayette, Indiana
mlohrum@purdue.edu

Abstract. The Nokia N900 is a very powerful smartphone and offers great utility to users. As smartphones contain a wealth of information about the user, including information about the user's contacts, communications, and activities, investigators must have at their disposal the best possible methods for extracting important data from smartphones. Unlike with other smartphones, knowledge of forensic acquisition from the N900 is extremely limited. Extractions of data from the N900 are categorized into limited triage extractions and full physical extractions. The imaging process of the phone has been explained as is necessary for a full investigation of the phone. The types of data as called for in a limited data extraction have been identified, and the locations of these files on the N900 were detailed. Also, a script was created which can be utilized for a limited data extraction from a Nokia N900.

Keywords: mobile forensics, smartphone forensics, Nokia N900, Maemo.

1 Introduction

The technology of communications by mobile devices has greatly advanced. Radio communications have evolved into car phones, cellular telephones, camera phones, and smartphones, the newest evolution of mobile devices. Smartphones have become ubiquitous, and there exists a great variety of manufacturers and models of these devices, along with various operating systems. The Nokia N900, running the Maemo operating system, is a very powerful phone and offers great potential in terms of utility to the user. In the community of Digital Forensics Sciences, little is known about acquiring data from the N900. This phone and the Maemo operating system are discussed in detail in this paper, along with the locations of data of importance to investigators, and methods of accessing important data.

The Nokia N900 is an extremely powerful device. Billed as a mobile computer, as opposed to a phone, this device is of great sophistication. A 600 megahertz processor is onboard, along with 256 megabytes of RAM and 768 megabytes of virtual memory. 32 gigabytes of persistent storage is included, along with a microSD slot allowing for up to 16 more gigabytes. The smartphone can connect to GSM networks and can handle 3G data transfer, and 802.11b/g WLAN support is built in. GPS capability is integrated into the hardware. And also, a camera is included, supporting still images at 5 megapixels with a flash, and video at up to 840 pixels by 480 pixels. It is the only smartphone in existence to run Maemo [12].

P. Gladyshev and M.K. Rogers (Eds.): ICDF2C 2011, LNICST 88, pp. 253–267, 2012.

Maemo is a Debian Linux-based operating system developed by Nokia optimized for mobile phones and internet devices. The current version of Maemo, and the topic of this study and paper, is Maemo 5, which is installed on the Nokia N900. Maemo 5 is the first to support for High Speed Packet Access (HSPA). Previous versions of Maemo were used on internet tablets [11].

Maemo is an extremely powerful and feature-heavy operating system. Such features include touch-screen interaction with multiple gestures; multi-tasking and task management; a browser supporting Adobe Flash and web history; phone functionality; messaging via SMS, MMS, Skype, and more; contact lists including sharing statuses and updates with others; and video and photo support for the camera [10].

Many people utilize the features on the phone aimed at advanced users. A Unix terminal is included, allowing Linux users to do advanced activities with the phone, including and not limited to file creating and movement, application downloading and customizing, and advanced networking. Also, the user is able to gain root, allowing functionality which standard users are not able to access. Users may also download applications to extend the functionality of the phone. A reviewer who had the opportunity to use this device performed many non-traditional phone tasks, including Skype and VOIP communications, using SSH, and playing old video games on NES and SNES emulators [3], [13].

1.1 Smartphone Forensics

Like any other technology, trends in mobile phones change. Smartphones have become extremely popular in recent years, and it is projected right now that 42% of Americans use smartphones [2]. To a forensic investigator, this figure means that smartphone forensics is an area where research is needed aimed at producing techniques applicable to acquiring evidence from these devices.

Before discussing smartphone forensics, it is best to define a smartphone. According to the National Institute of Standards and Technology [9], a smartphone is "[a] full-featured mobile phone that provides users with personal computer like functionality by incorporating PIM applications, enhanced Internet connectivity and email operating over and Operating System supported by accelerated processing and larger storage capacity compared with present cellular phones". And according to the same source, Personal Information Management (PIM) is defined as "Data that contains personal information, such as: calendar entries, to-do lists, memos, reminders, etc."

Though smartphone forensics is an area where more research is required, a good amount of knowledge exists regarding forensic extractions from many smartphones. Forensic extractions from iPhones, Android powered phones, and Symbian powered phones are documented academically [4], [5], [7], [14].

As is, there is minimal academic knowledge of Nokia N900 forensics. There is one web page [1] which discusses locations of evidence from the N900. There is some useful information on this page, but the information has not been verified, and little discussion is offered on the significance of the findings. The author of the page asks for viewers to share information of significance on this phone, so there is

definitely an understanding that forensic extractions from the N900 is in its infancy. Given the knowledge of forensics of other phones and smartphone operating systems, and given the potential of data hiding in the Maemo operating system, it seems that N900 forensics should be an area of research.

An investigative model exists to aid investigators in rapidly examining mobile phones because phones are so prevalent. Mislan, Casey, and Kessler [6] define a set of functions of triage investigations of mobile phones, including such purposes as finding evidence for a case, assessing the severity of the crime and the danger of the criminal, determining if there are any victims at risk, and deciding if further investigation, such as a full examination of the phone's internal persistent storage media, is warranted. The authors also propose a six step guide for triage investigations of mobile devices. The six steps are to initiate chain of custody, isolate the device from the network if possible, disable security features if possible, limited triage data extraction, reviewing triage data, and previewing removable storage media. The step of limited triage data extraction is the primary focus of this paper, as this step involves locating and extracting data stored on the phone. Entries that would qualify as limited triage data extraction would generally include user data, such as call logs; text messages, contacts, and calendar entries; files, such as videos and photographs; and device identification information, such as an International Mobile Equipment Identity (IMEI) number. This data aids in explaining about who a person knows, which people are important enough to keep as contacts and as quick dials, who the person has been in contact with recently, what the person has been up to recently, and what the person may be planning to do in the future.

Once a triage examination is performed, the investigator may choose to further examine the phone. If this is the decision, securely moving the phone back to a laboratory environment is the next step. If possible, the investigator would want to obtain a physical image of the entirety of the phone's memory. However, depending on what kind of phone is being examined, the process may entail accessing memory via a JTAG interface or even desoldering and removing a chip from the phone itself. Obtaining a physical image of a phone, like a computer, allows access to data in unallocated space, such as deleted data [8].

2 Research Methodology

The goal of research is to find the locations of evidence detailed above on the Nokia N900. It is prudent to understand how data is stored in files in these locations and also to ensure accuracy of this data.

The researcher used a Nokia N900 for a period of seven days and completed normal activities, including placing and receiving phone calls, sending and receiving text messages, adding calendar events, taking photographs and videos, and web browsing. As much activity as possible was logged by hand on paper sheets, including times and elapsed time of events and phone numbers involved in conversations; it was not possible to log all activities, as it was difficult to have access to these sheets at all times and was distracting in some occasions to be logging activity, such as while driving.

After a seven day period, the researcher created a full physical image of the phone using a method to be described in the following sections and explored the image for locations of relevant data, using Access Data's FTK Imager and Forensic Toolkit 1.81.6. The researcher noted the locations of this data and any issues involving locating, copying, or parsing through this data. Upon finding this data, the researcher compared the logged data with the data recovered from the phone in order to understand what all entries in the phone correspond to with the written logs. Finally, after finding the locations of data, the researcher performed a limited triage extraction onto a microSD card to prove that such a process can be done.

2.1 Putting the Phone into Offline Mode

As stated previously, a step in a triage investigation of a mobile phone is to isolate the device from network, if feasible and applicable. Removing a phone from the network is also useful for other purposes, such as flying on an airplane. Also, when the device's wireless capabilities are enabled but the network is undetectable, such as on a subway, the device devotes more power to finding a signal, and this drains the battery. Disabling the network connections allows the device's functionalities, outside of wireless capabilities, to be used in these situations without draining the battery. On the Nokia N900, there is a button on the center of the top wall of the case. Clicking this button brings a menu, including a button for offline mode. This turns off all wireless capabilities, which isolates the phone from the network.

2.2 Physical Image of the Phone

The researcher created a physical image of the phone and explored the image for files of importance. Creating a physical image allows an investigator to view all contents of the phone, including remnants of deleted files, for evidence; exploring the physical image is far quicker than exploring the phone itself as exploring the phone is limited to the use of command line utilities, where exploring the image of the phone allows the use of Access Data's FTK Imager. Unlike most phones, it is very easy to obtain a physical image of the Nokia N900. Maemo allows the dd command to be utilized. Dd allows for a physical image to be made of the entire phone or a single partition, and it also allows for other commands, such as file copying, file erasing, and zeroing out a partition.

This process requires a micro USB cable and a computer running a Linux build, such as Ubuntu, to connect to the phone. A usable option is to create a virtual machine running Ubuntu instead of using a full Ubuntu machine.

Creating a dd image requires two pieces of software to be on the phone, Rootsh and openSSH Host. It is possible that they already are installed. If not, the investigator must download and install these applications from the Application Manager. OpenSSH allows the phone to be an SSH server, and Rootsh allows root access to the phone. The image will be created and transferred over a USB cable via SSH to an Ubuntu computer.

It is necessary to know the root password of the phone. Installing Rootsh and openSSH allows the investigator to set the root password. If these are already installed and the investigator does not know the root password, the investigator can open the terminal program and type the following lines:

```
sudo gainroot
passwd
```

These lines gain root access, then allow the root password to be changed, and entering the current root password is not required to enter a new root password.

The investigator now should connect the phone to the Ubuntu computer via USB and choose to mount the phone in Phone Suite mode. At this point, the investigator should open the Terminal application on the phone and type the following:

```
sudo gainroot
ifconfig usb0 192.168.99.1
```

These commands gain root access, which is required for creating an image, and then create a network connection over USB to the computer and gives the phone the address 192.168.99.1. Next, on the Ubuntu computer, the investigator should type the following:

```
sudo ifconfig usb0 192.168.99.2
ssh root@192.168.99.1 cat /dev/mmcblk0 > home/[Ubuntu
user name]/mmcblk0.img
```

The first command completes the network connection over USB and gives the computer the IP address of 192.168.99.2. The second command creates the file mmcblk0.img on the home folder of the computer, then redirects the contents of mmcblk0 in the phone's dev folder to this file. This line effectively creates a dd image of the entirety of the phone's memory on the computer.

The folder /dev on Unix machines stands for device and contains symbolic links to important blocks in memory. Mmcblk0 points to the first block on the phone; running the dd command on this block creates a physical image of all three partitions. It is also possible to create images of individual partitions. Mmcblk0p2 refers to the second partition, which is where the operating system resides. This can be accomplished by the same commands as above, except replacing mmcblk0 with mmcblk0p2. Also, because the second partition is only 2 gigabytes, this can be transferred to a microSD card if the card is big enough and has enough space. This method does not require downloading and installing openSSH. On the phone, the investigator can type the following to image the second partition in this fashion:

```
dd if=/dev/mmcblk0p2 of=/media/mmc1/mmcblk0p2.img
```

2.3 Persistent Storage

Once the physical image was created, the researcher could fully examine the details of the phone's persistent storage. The Nokia N900 has nearly 32 gigabytes of persistent storage; that is a lot of storage for potential evidence which must be explored. As the Maemo operating system is based on Unix, much is already known about the structure of the phone, but there still is much to be learned as this is a modified version of the OS optimized for mobile phones.

The N900 storage memory is divided into three partitions: 27652 megabytes, or nearly 28 gigabytes, formatted as FAT32 intended for user storage, 2 gigabytes formatted as ext where the operating system and phone data is stored, and 768 megabytes of unformatted swap space.

2.4 Triage Extraction of the N900

As explained previously, obtaining a full image of the phone is easy; this creates a great impact on forensic extractions from the N900. A full physical image can be created without taking apart the phone and accessing the memory chips, and the investigator need not purchase expensive mobile forensics software. When investigating a hard drive, it is common practice to obtain a full physical image and examine it in the lab. This can also be done with the N900, allowing traditional examinations, like with hard drives, to be performed on the phone, including data carving for extracting deleted files. And as Unix forensics is relatively well understood, and because the phone has such an amount of storage and such potential to contain evidence similar to be found on a hard drive, the focus of data examined in this paper will be on triage investigations and data sought after when time is of the essence. It is recommended that an on-scene triage investigation is performed, which involves the investigator inserting a clean microSD card to the phone and copying important files to this card. Next, the investigator securely moves the phone, and the suspect's microSD card if acquired with the phone, to a digital forensics laboratory, then obtains a full physical image and thoroughly examines the image of the phone, and the image of the card if applicable.

It is important, therefore, to define what data is to be sought after in a triage examination of an N900. As stated previously, the data sought after in general in these styles of investigations includes call logs, text messages, contacts, calendar entries, videos, photographs, and an IMEI number [6]. Because of the nature of Maemo and its Internet capabilities, it is also important to add web artifacts, including browsing history, typed URLs, and bookmarked pages.

3 Results

The following sections explain the results of completing the research. The files found are presented, their locations are documented, and the significance is explained. Also, a script is presented which can be used to extract all of these important files to an SD card in the phone.

3.1 File Locations

With a physical image of the phone created, the researcher was able to explore the image instead of exploring the phone for relevant files. Knowing how to find a file in a physical image, though, can be different from knowing where to find the file on the phone itself. It is important to differentiate between a file's location in terms of file system location and in terms of logical location. A file system location refers to a file's location within the file system when the system is mounted on another system. This allows a user to explore files as they are stored, as opposed to how they are presented. A file's location logically refers to where the file is located when the user is using the phone and the operating system, as such are presented to the user. Not all areas of the phone are available to the user, so some files may be located in positions other than their file system locations. For example, the following root folders are found in the file system layout in the following partitions:

Table 1. File system root folders

Partition	1	2
File System	FAT32	Ext3
Root Folders	.apt-archive-cache .documents .images .sounds .videos Cities DCIM Mac OS	lost+found Opt User

However, the following folders are on the logical root. This is determined by typing the following in the terminal:

```
sudo gainroot
cd /
ls -A
```

Table 2. Logical root folders

.dev	Dev	initrd	opt	srv	usr
Bin	Etc	lib	proc	sys	var
Boot	Floppy	media	root	syspart	
Cdrom	Home	mnt	sbin	tmp	

Both the file system and logical locations are noted in the following files of forensic interest, along with an annotation and explanations of data headers, if necessary.

3.2 Browser Artifacts

File system location: Partition 2/user/.browser_typed_urls
Logical location: /home/user/.browser_typed_urls

This file is a flat file that contains all URLs typed by the user. It does not contain dates or any other data; it only contains the URLs as the user typed them.

File system location: Partition 2/user/.mozilla/microb/places.sqlite
Logical location: /home/user/.mozilla/microb/places.sqlite

This is a SQLite database file which holds history from the built-in web browser, which is just a Mozilla browser. Within this database is a table called moz_places, which serves as web history. Columns of interest are as follows:

Table 3. places.sqlite, Table moz_places

Column	Description
url	This is the url that was visited.
Title	This entry holds the name of the page that was visited.
visit_count	The number here is how many times the page was visited.
last_visit_date	Stored in Unix epoch time, the entry here is the last time the page was visited.

File system location: Partition 2/user/.mozilla/microb/cookies.sqlite
Logical location: /home/user/.mozilla/microb/cookies.sqlite

This is a SQLite database file which holds cookies from the built-in web browser. Cookies can aid in discovering information about the user's browsing behavior. There is one table within this, called moz_cookies. Columns of interest are as follows:

Table 4. cookies.sqlite, Table moz_cookies

Column	Description
Name	This holds the name of the cookie and may or may not be useful.
Value	This holds the value of the cookie.
Host	This is the host where the cookie was generated, which implies that a user visited that host.
Expiry	Stored in Unix Epoch time, this holds the date and time when a cookie expires.
lastAccessed	Stored in Unix Epoch time, this holds the date and time a cookie was last accessed.

File system location: Partition 2/user/.mozilla/microb/signons.sqlite
Logical location: /home/user/.mozilla/microb/signons.sqlite

This is a SQLite database file which holds information about automatic sign-ons from the built-in web browser. Within this database is a table called moz_logins. Though the username and password are both encrypted in this table, having a login saved means the page on the table is of great interest to the user. Columns of interest are as follows:

Table 5. signons.sqlite, Table moz_logins

Column	Description
Hostname	This holds the hostname of the login.
formSubmitURL	This holds the specific URL of the login.
usernameField	This is the ID for the textbox on the page which holds the username.
passwordField	This is the ID for the textbox on the page which holds the password.
encryptedUsername	This is encrypted, so the username cannot be directly found from this entry.
encryptedPassword	This is encrypted, so the password cannot be directly found from this entry.

Within the same database file is a table called moz_disabledHosts. This appears to contain URLs which the user was prompted to save a login and chose not to save login credentials.

3.3 Phone Logs and Text Messages

File system location: Partition 2/user/.rtcom-eventlogger/el-v1.db
Logical location: /home/user/.rtcom-eventlogger/el-v1.db

This file is a SQLite Database file with a table called Events which tracks all telephone-related events, including phone calls, missed calls, and text messages. Columns of interest are as follows:

Table 6. el-v1.db, Table Events

Column	Description
Event_type_id	This associates with another table in the database called EventTypes. Events include calls, missed calls, SMS messages, and others.

Table 6. (*continued*)

Storage_time Start_time End_time	These three times may or may not be included in each record. They will all be within a few seconds of each other. The researcher was hoping to be able to determine call lengths by subtracting start_time from end_time in call records, but the only time recorded in both texts and calls is the beginning of the communication. Times are stored in Unix epoch time.
Is_read	If the entry is a text message, a 1 is entered here if the message has been read. Otherwise, even if the entry is a call, a 0 is entered.
Outgoing	Whether the entry is a text or a phone call, a 1 will be placed if the event is an outgoing event. If it is an incoming event, a 0 is entered.
Remote_uid	This is the phone number on the other end of the communication.
Free_text	If the entry is a text message, this is the contents of the message. Otherwise, this is blank.
Group_uid	If the entry is a text message, this is a group identification used so that conversations can be grouped together. Otherwise, this is blank.

3.4 Address Book

File system location: Partition 2/user/.osso-abook/db/addressbook.db
Logical location: /home/user/.osso-abook/db/addressbook.db

This file stores the address book in the VCard format, and this stores the date and time of entry, phone number, and name of each contact. This file can be very easily read in Notepad or a hex editor as all entries are stored in plain text and are easy to interpret.

3.5 Calendar

File system location: Partition 2/user/.calendar/calendardb
Logical location: /home/user/.calendar/calendardb

This is a SQLite database file which holds entries in the calendar as created by the user. Within this database file is a table called Components, which holds entries that related directly to calendar events. Columns of interest are as follows:

Table 7. calendardb, Table Components

Column	Description
Flags	Flags
DateStart	Stored in Unix Epoch time, this holds the date and time when an event starts
DateEnd	Also stored in Unix Epoch time, this holds the date and time when an event ends
Summary	This is the title of the event
Location	This is the location of the event
Description	This represents details entered by the user about the event
Status	Unknown. Integer representing some sort of completion status.
Until	Unknown. Unix epoch time, possibly representing when a repeating event is to end.
AllDay	This represents if an event is all day or not. 0 represents that an event is not all day
CreatedTime	Stored in Unix Epoch time, this entry holds when an event was created.
ModifiedTime	Stored in Unix Epoch time, this entry represents a time when an event was edited. It appears a user deactivating an alarm reminding of an event constitutes editing.
Tzid	Either the time zone of the event or the time zone where the event was entered.
TzOffset	The offset of the time zone of Tzid in minutes.

3.6 Multimedia Files

Videos and pictures are stored on the main FAT32 partition. These are located by default in the same folder on the first partition under the root folder DCIM. Logically, they are located at ./home/user/MyDocs/DCIM. In both photos and videos,

the date is included by default in the filename, and the created date and time refers to the time the picture or video was taken.

3.7 E-Mail

The researcher linked the phone to a personal GMail account. There are many locations for e-mail artifacts. In the file system, e-mail artifacts are stored under /user/.modest/cache/mail/imap, and in that directory there was a folder with the e-mail address in the folder name. It is assumed that within /user/.modest/cache/mail could be other folders named for e-mail protocols, such as pop3, and other e-mail folders within said protocols. Also, there was a directory in the file system /user/.modest/local_folders/sent which contained e-mail sent from the phone. There also existed a folder /user/modest/local_folders/drafts, though this was empty. It is assumable that e-mail drafts that weren't sent would be stored here. More research is required into how the phone stores e-mails from different account types.

3.8 IMEI

There does not appear to be a file in the phone containing the IMEI number. The number appears in the third partition, which is unformatted swap space. This implies that the IMEI number is not stored in the phone's storage but is saved on another chip on the phone. It likely appeared in swap space because the number appeared on a screen visible to the user, then was saved in swap space temporarily. However, the IMEI number can be obtained by interacting with the phone. Clicking on Applications -> Settings ->About Product reveals the IMEI number.

3.9 Simple Triage Extraction

Now that evidence as described for mobile phone triage investigation has been located, the method of conducting a triage investigation on the N900 is described. First, the investigator should secure the phone, note the state of the phone, note if it is powered on or off, take a photograph of the screen, and note any details of the phone, such as cracks, scratches, materials on the phone, markings on the phone, where it was found, and who owns it. If the phone is on, the investigator should remove the back cover and see if there is a microSD card. If so, this card should be removed and secured properly, and a trusted microSD card should be now inserted which can contain triage data from the phone. If the phone is off and the investigator is qualified to investigate a phone, turning the phone on and completing a triage investigation is acceptable; otherwise, keep the phone off. However, before powering the phone on, the investigator should remove the back cover and check for a microSD card as stated before, insert one for completing the triage investigation, then power the phone on and complete a simple triage extraction. It is important to note that a microSD card will not mount if the back cover of the phone is not attached.

Using the locations as documented previously, the following lines can copy files from the phone to the microSD card:
Copy typed urls:
cp /home/user/.browser_typed_urls /media/mmc1/.browser_typed_urls
Copy Firefox browsing history:
cp /home/user/.mozilla/microb/places.sqlite /media/mmc1/places.sqlite
Copy Firefox cookies:
cp /home/user/.mozilla/microb/places.sqlite /media/mmc1/cookies.sqlite
Copy Firefox sign-ons:
cp /home/user/.mozilla/microb/places.sqlite /media/mmc1/signons.sqlite
Copy call and text event history:
cp /home/user/.rtcom-eventlogger/el-v1.db /media/mmc1/el-v1.db
Copy address book:
cp /home/user/.osso-abook/db/addressbook.db /media/mmc1/addresssbook.db
Copy user calendar:
cp /home/user/.calendar/calendardb /media/mmc1/calendardb
Copy videos and pictures:
cp –r /home/user/MyDocs/DCIM /media/mmc1/DCIM
Copy e-mail artifacts:
cp –r /home/user/.modest /media/mmc1/modest

It is possible to script the above commands. A script in a Unix operating system, like Maemo, allows functionality to be completed automatically. A script can be created called N900TriageExtraction.sh, and executed by entering into the following command into the phone:

```
./N900TriageExtraction.sh
```

The script file, TriageExtraction.sh, should be written as follows:

```
#!/bin/sh
# Copy typed urls:
cp /home/user/.browser_typed_urls
/media/mmc1/.browser_typed_urls
# Copy Firefox browsing history:
cp /home/user/.mozilla/microb/places.sqlite
/media/mmc1/places.sqlite
# Copy Firefox cookies:
cp /home/user/.mozilla/microb/places.sqlite
/media/mmc1/cookies.sqlite
# Copy Firefox sign-ons:
cp /home/user/.mozilla/microb/places.sqlite
/media/mmc1/signons.sqlite
```

```
# Copy call and text event history:
cp /home/user/.rtcom-eventlogger/el-v1.db
/media/mmc1/el-v1.db
# Copy address book:
cp /home/user/.osso-abook/db/addressbook.db
/media/mmc1/addresssbook.db
# Copy user calendar:
cp /home/user/.calendar/calendardb
/media/mmc1/calendardb
# Copy videos and pictures:
cp –r /home/user/MyDocs/DCIM /media/mmc1/DCIM
# Copy e-mail artifacts:
cp –r /home/user/.modest /media/mmc1/modest
echo "Triage extraction completed."
```

Once the script is completed, all of the files as discussed will be on the SD Card's root. At this point, a basic extraction has been completed. Next, the investigator should securely transport the phone back to a laboratory for a full examination, including creating a full physical image of the phone and examining the contents, treating the image as both an image of a phone and an image of a Unix computer.

4 Discussion

The locations of much information have been documented on the Nokia N900, including browser artifacts, phone and text events, calendar events, multimedia files, and e-mail artifacts. Though much information has been found, it would be nice to also find where bookmarks from the web browser are stored, as bookmarks show that a user had great interest in a webpage. The user has the ability to save bookmarks to the phone's main menu. Though the data locations of these artifacts weren't found either, an examiner can note what links appear on the main menu. Among the data locations documented by Bryner [1] were the address book, call and text message logs, and browser artifacts. As the data locations in that webpage and in this paper are identical, it is likely that the other data locations in that webpage are quite accurate. It is recommended that an examiner studying N900 thoroughly in a lab environment also use Bryner's webpage as a reference.

The method of obtaining a physical image discussed in this paper may raise questions to experts in the area of digital forensics. This method requires installing the RootSH and openSSH programs. These programs are small and leave a minimal footprint on the phone. According to the application manager, the download size of Rootsh is 1 kb, and the application requires 32 kb of disk space on the operating system partition. OpenSSH Server's download size is 261 kb, and it requires 656 kb of space. On a 2 gigabyte partition, this is a very small footprint.

The Nokia N900 and the Maemo operating system clearly can hold a lot of information about a user. This information varies from personal management

information, contacts, multimedia, internet activities, and more. Because time is often of the essence in investigations, directions to complete an on-scene extraction of information from an N900 phone have been presented. The more knowledge of rapid triage extraction from mobile phones that exists in the academic community, the better prepared investigators will be in the future when on-scene results are required.

Acknowledgments. The author would like to thank Rick Mislan, Assistant Professor of Computer and Information Technology at Purdue University, for guidance on this paper and in the area of smartphone forensics. Mislan is an instructor and advisor in Cyber Forensics, focusing on Small Scale Digital Device Forensics, and is the author's graduate school advisor.

References

1. Bryner, J.: Nokia N900 Mobile Forensic Cheat Sheet (2010), http://blogs.sans.org/computer-forensics/2010/03/17/nokia-n900-forensic-cheat-sheet
2. Dolan, B.: Study: 42 percent of U.S. Uses a Smartphone (2010), http://mobihealthnews.com/6178/study-42-percent-of-u-s-uses-a-smartphone
3. Evans, M.: Hands-on Nokia N900 Review – The Best Nokia Smartphone Yet (2009), http://mobilementalism.com/2009/09/14/hands-on-nokia-n900-review-the-best-nokia-smartphone-yet
4. Hoog, A.: Android Forensics (2009), http://viaforensics.com/wpinstall/wp-content/uploads/2009/08/Android-Forensics-Andrew-Hoog-viaForensics.pdf
5. Hoog, A., Strzempka, A.: iPhone Forensics White Paper (2009), http://viaforensics.com/education/white-papers/iphone-forensics
6. Mislan, R.P., Casey, E., Kessler, G.C.: The Growing Need for On-Scene Triage of Mobile Devices. Digital Investigation 6(3-4) (2010)
7. Mokhonoana, P.M., Olivier, M.S.: Acquisition of a Symbian Smart Phone's Content With An On-Phone Forensic Tool. In: Proceedings of the Southern African Telecommunication Networks and Applications Conference (2007)
8. Mooij, B.: Data Extraction From A Physical Dump, http://www.dfinews.com/article/data-extraction-physical-dump
9. National Institute of Standards and Technology.: Smart Phone Tool Specification, Version 1.1 (2010), http://www.cftt.nist.gov/documents/Smart_Phone_Tool_Specification.pdf
10. Nokia Corporation.: Maemo Features, http://maemo.nokia.com/features
11. Nokia Corporation.: Stories Behind Maemo: Bringing Open Source to the Consumer Mainstream, http://maemo.nokia.com/maemo/open-source
12. Nokia Corporation.: Technical Specifications, http://maemo.nokia.com/n900/specifications
13. Rankin, K.: Nokia N900: First Look (2009), http://www.linuxjournal.com/content/nokia-n900-first-look
14. Zdziarski, J.: iPhone Insecurity, http://iphoneinsecurity.com