INTERNATIONAL JOURNAL OF ENHANCED RESEARCH IN MANAGEMENT AND COMPUTER APPLICATIONS
VOL. 2, ISSUE 1, JAN. 2013 ISSN NO: 2319-7471

Efficient Generalized Forensics Framework for extraction and documentation of evidence from mobile devices

Rizwan Ahmed[1], Dr. Rajiv V. Dharaskar[2], Dr. Vilas M. Thakare[3]
[1]Department of Computer Science and Engineering, G. H. Raisoni College of Engineering, Nagpur, India.
[1] rizwanmailbox@gmail.com

Abstract: The Google's Android mobile platform is the most popular mobile operating system in terms of shipment of devices [21]. The Android platform provided significant advantages for consumers with respect to competition and features. Due to lack of knowledge and supported tools for investigating Android powered devices, the forensic investigators have struggled [1, 2]. In this paper, we present the efficient generalized forensics framework for acquisition and subsequent analysis of these devices.

Keywords: Android, Digital Evidence, Mobile Forensics, Mobile Forensic tools, Smartphone.

I. INTRODUCTION

According to Nielsen January 2012 survey, 46.3 percent of all smartphone owners have an Android device. But, 51.7% of recent acquirers of new smartphones have chosen Android devices over Apple iPhone [27].

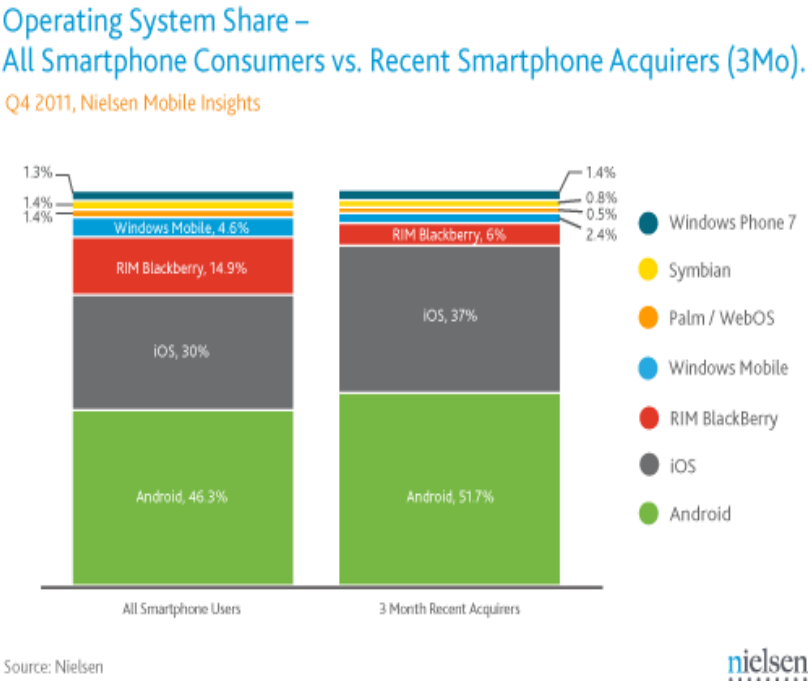

Figure 1. Operating System Share- Nielsen 201.1

Although most of the discussed statistics about Android focus on smartphones and now tablets, there are many more devices that currently or in the near future will run on Android. Some examples include vehicles, televisions, GPS, gaming devices, netbooks, and a wide variety of other consumer devices [1].

These smartphone devices are getting more and more sophisticated in terms of processing power and available features making them equitable to modern PC's. But, this is also posing important security risk of these devices being used for carrying out digital crimes or being target of a

security attack due to predominant use by employees at various enterprises. IT firm IBM has warned that malware targeting mobile devices is on the rise with the tripling of critical vulnerabilities this year compared to last year. The IBM report cited the G Data Security Labs' findings that the number of smartphone and tablet malware increased 273 percent in the first semester of 2011 compared to the same period last year [3].

As Android devices grow in numbers, an increased awareness of the data they possess will equally grow. Unfortunately, much of that interest will come from cyber criminal organizations who realized that successful attacks against the platform will yield significant results as the devices contain enormous quantities of personal and business information [1]. Android devices have also been vulnerable to various kinds of security and malware attacks [4]. According to McAfee in their new study, the number of viruses, trojans, and other rogue pieces of code aimed at Google's Android platform shot up 76 percent [5]. Lots of similar security vulnerability reports keep coming about Android platform almost regularly now [6, 7, 8].

However forensic analysis and security engineers have struggled as there is lack of knowledge and supported tools for investigating these devices [1, 2]. This paper tries to analyze issues not only providing in-depth insights into Android hardware, software and files system but also by studying techniques for the forensic acquisition and subsequent analysis of these devices.

II. RELATED WORK

Currently, numbers of researchers had addressed to the security issues of the smartphone, and developed various technologies for the investigative features. In this section, we have analyzed the definitions of digital evidence, mobile forensics and smartphone, and also introduced some studies that had down in Android smart phone operating system architectures, and mobile phone forensic tools area.

A. Digital Evidence

The digital evidence is a series of binary digit numbers on transmission [9], or stored information files on the electronic device. Moreover, the digital evidence file formats includes audio, video, images, and digital, etc. The digital evidence is not virtual exist, but there are some other features to look for, the digital evidence can be copied with unlimited differences, can be modified easily, hard to be identified the original resource, can be integrated data verification, and cannot be understood directly without technical process.

B. Mobile Forensics

With the increased emphasis on social security issue, crime issue is considerable when it comes to the utilization of smart phone technologies, digital forensics provide the technical skills to collect evidences for the court to review and judge cases. Digital equipment has changed daily, people has pervasive use some common digital devices such as computers, Internet, mobile phones, digital cameras, hardware, storage devices, etc. Currently, digital forensics has widely used in the areas of network forensics, mobile forensics, computer forensics, and memory forensics, etc. According to NIST definition of mobile phone forensics process is preservation, acquisition, examination and analysis, and then reporting [10].

The various aspects of mobile forensics have been discussed in our previous research work [11, 12, 13, 29, 30].

C. Smartphone

Due to the advanced technological development, mobile phone's selling was decreased in 2009; smart phones' selling is increased, and the commercial demand cannot be sacrificed by the smart phone. In Table 1 [14] shows definition of smart phone, the various categories of smart phones' forensic, different operating systems and the disordered domestic laws for forensic procedures result in the difficulty of smart phone forensics [15].

Table 1. Definition of Smart Phone

Item	Definition
Capability	With voice and data wireless communication personal management (PIM), such as contacts, calendar, alarm clock, etc.
Input Mode	Common with push-button, voice input, touch and multi-touch
Wireless Transmission	IrDA, Bluetooth and Wi-Fi
Operating System	Symbian, iphone, Windows Mobile, Android, Palm, RIM, etc.
Processor	Embedded multi-task microprocessor

III. BACKGROUND: ANDROID AND MOBILE FORENSICS

Google's Android is an open source smart phone operating system, which is based on Linux.

A. History of Android

A central figure in the development of Android is Andy Rubin and his company "Android, Inc" formed in 2003 which was subsequently acquired by Google in July 2005 [1].

On November 5th 2007, Andy Rubin announced Android as an open and comprehensive platform for mobile devices to be further developed by "Open Handset Alliance" comprising of more than 30 technology and mobile leaders including Motorola, Qualcomm, HTC and T-Mobile [16]. In 2007, Google released an early look at the Android software development kit (SDK) to developers followed by first Android Developer Challenge. The top 50 apps are available for review here [17].

B. Android OS

Android's kernel is a fork of the Linux kernel but has further architecture changes by Google outside the typical Linux kernel development cycle [18]. For example Android does not have a native X Window System nor does it support the full set of standard GNU libraries, and this makes it difficult to port existing Linux applications or libraries to Android [19]. The open strategy behind Android naturally let to the release of Android source code through AOSP on October 21, 2008 [20].

C. Android Architecture

Android is composed by five major components [22], depicted in Figure 2, that are briefly introduced below:

Figure 2. Android Architechture.

- Applications:
  Android is distributed with a set of typical applications for Mobile devices (e.g., e-mail client, text messaging management, browser, contacts management) written using the Java Programming Language.
- Application Framework:
  Android offers the capability of Java applications development providing a rich set of services which can be exploited. Developers can consume and provide services through of a wide set of Application Programming Interfaces (APIs), with the objective of the reuse of components, always respecting the security constraints enforced by the framework.
- Libraries:
  Android includes a set of libraries (e.g., Standard C System Library, Media Libraries, 3D Libraries) used by the components of the system through the Android Application Framework just outlined.
- Android Runtime:
  The Runtime is composed by a set of Core Libraries and by the Dalvik Virtual Machine (DVM). Every running application holds its own instance of the DVM and executes in its own process.
- Linux Kernel:
  One of the most interesting features of Android is the underlying Linux kernel supporting the core services, such as memory and process management, network stack, drivers and security.

D. Overview of Android File System

Another interesting element of Android is the natively supported YAFFS2 File System (FS). YAFFS [23] stands for Yet Another Flash FS and, at the time of writing, it is the only FS that has been specifically designed for NAND flash chips. The use of NAND flash chips in the field of embedded and mobile devices [24] is increasing and replacing the common-old NOR chips because of the improved density, speed and the reduced cost. At the time of writing, YAFFS was released in two version: YAFFS1: designed for old NAND chips with 512 byte pages plus 16 byte spare areas; YAFFS2: evolved from YAFFS1 to accommodate newer chips with 2048 byte pages plus 64 bytes spare areas.

In addition to the different NAND chips supported, YAFFS2 evolved in terms of performance, reliability, efficiency and support to the "write once" requirement for modern NAND flash [24, 25].

In the next section, a brief description of some items related to the Android Security Architecture is provided; a more extended description is available at Android security architecture [26].

E. Android Security Architecture

Android is a multi-process platform which relies on the standard Linux facilities for processes and users management; in fact, most of the security between applications is enforced at process level exploiting such standard facilities. However, in order to support the reuse of components and the provisioning of services between different applications, some finer-grained security features are provided by the mechanism of permissions.

- **Applications and sandboxes:**
  Android, by default, denies to any application the capability to perform operations with the objective to hamper any other application, the OS or the end-user. Hence, due to this design pillar, for applications it is impossible to perform any operation on end-user private data (e.g., contacts, messages), to gain access to another application's files, to perform network accesses, to manage the device state, and so on. Following this idea, Android binds any running application to a secure Sandbox which cannot interfere with any other applications, except by the explicit declaration of the required permissions to access to the desired capabilities which are not provided by the Sandbox. The set of permissions held by an application is defined in a static way, verified at installation time and cannot change during the lifetime of the application. Any Android application is required to be signed with a certificate, held by the developer, in order to establish and to manage relationships between applications.

- **User IDs and permissions:**
  By default, Android manages each installed application as a different Linux user; in fact, at installation time, any application is provided with its own unique Linux user ID. All the data stored by a given application will receive the application's user ID as well; to grant to other applications any access to such data, it is required to enable the access from the Others group of Users

  By default, a basic Android application has no associated permissions; in order to overcome the limitations which could arise using only the DVM default capabilities, and to allow service provisioning between applications, it is possible to declare further permissions. The declaration of the needed permissions is performed at development time through the inclusion of <uses-permission> tags in the application's Android manifest.xml file. During the installation, the permissions required by the application are granted by the package installer module; the policy to grant permission can leverage both on applications' signatures and on interaction with the end-user. Once the application is installed, the set of the granted and denied permissions is built and cannot be modified: during the execution, no more checks are performed.

F. Android: Important Data From Forensics Perspective

Following data can be considered as important from forensics perspective on Android devices:

- Subscriber & equipment identifiers
- Date/time of calls, movements, etc
- Phonebook
- Appointment Calendar
- SMS, Text Messages
- Dialed, incoming, & missed call log
- Electronic mail
- Photos
- Audio and video records
- Multi-media messages
- Instant messages
- Electronic Documents
- Location information

IV. EFFICIENT GENERALIZED FORENSICS FRAMEWORK FOR EXTRACTION AND DOCUMENTATION OF EVIDENCE FROM ANDROID DEVICES

This section outlines our methodology for extraction of digital evidence from Android devices:

A. Acquisition Methodology

Figure 3. Acquisition Methodology

The approach we propose in this paper focuses on acquiring data from Android device's internal storage memory, copying data to an external removable memory card (like SD, min SD, etc.,) as shown in Figure 3. This task of forensic acquisition of evidence can be thus performed without need for connecting the Android device to PC. This will result in redeeming forensic operators to travel with luggage containing plenty of one-on-one tools for every single Android device.

B. Open Architechture

The following Figure 4 below shows the proposed architecture of Efficient Generalized Forensics Framework for Mobile Devices:

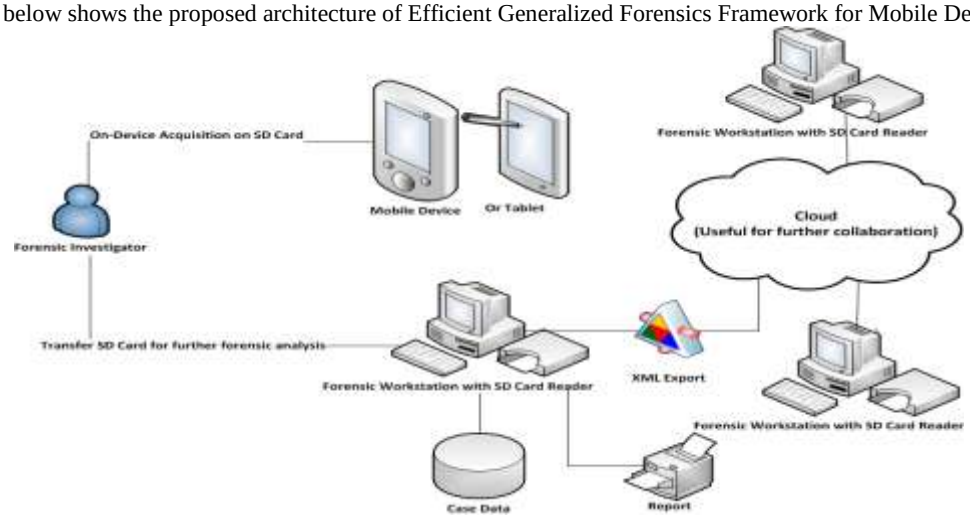

Figure 4. Open Architecture

In order to acquire data from Android devices all the following components will play very vital role:

- On-Device Acquisition on SD Card
- Forensic Workstation with SD Card Reader
- Case Database
- Case Reporting Module
- Open Architecture to Collaborate with other Forensic Workstation

C. Acquisition Process

The complete data acquisition process is shown in Figure 5 below:

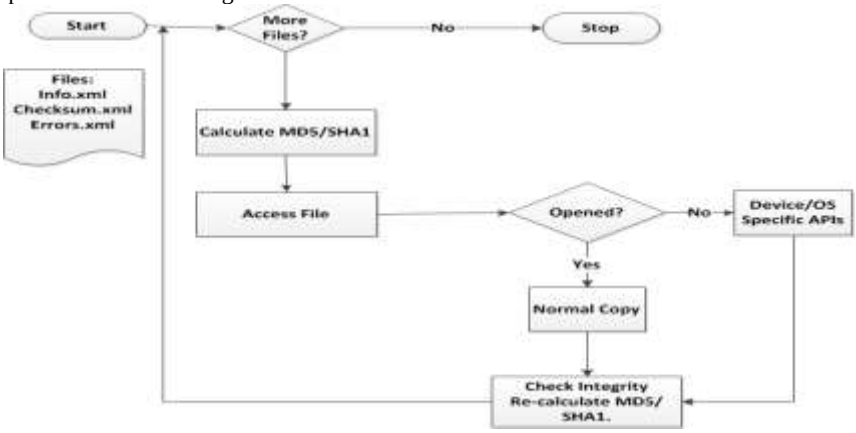

Figure 5. Acquisition Process

Before Acquisition process starts, it is necessary to shield the device with Farady cage to avoid network communication which could trigger events resulting in modification of file system's object. Mostly all the Android devices have option to plug-in a SD card while the device is powered-on (hotplug) without removing battery. This is really helpful since for collecting data which otherwise could be altered if the device is turned off before the seizure process. Therefore, we have to check first if a SD card is already plugged, and replace it with a SD card containing updated version of Efficient Generalized Forensics Framework Acquisition App. We need to then navigate through File Explorer to launch the Acquisition App. The App will kill all non-necessary processes running on the system in order to avoid locking problems. In order to insure integrity acquired data, the App performs hashing of each file before and after copy. The relevant information about all the file hashes are saved in Checksum.xml log file for further analysis later. Data acquistion is done using Device/OS specific API's along with deleted data using file allocation table.

D. Acquisition Algorithm

The implementation details are provided in the following Figure 6 which shows the pseudo-code for the Acquisition Process:

```
Algorithm Acquisition
    Input: A path p.
    Output: none.
    for all objects obj (files and directories) in p do
        if obj is a directory then
                Create a directory named p in the SD Card
                Recursively call Acquisition(p/obj)
        else if obj is a file then
                Compute MD5/SHA1 hash of obj
                Copy obj in path p on the SD Card
                If obj has not been copied then
                        Access to obj with Device Specific APIs
                        If obj could be accessed then
                                Recreate a similar database in path p on the SD Card
                        end if
                end if
    end if
    Compute MD5/SHA1 hash of the copied obj on the SD Card
    end if
    end for
```

Figure 6. Acquisition Algorithm

The above algorithm performs following two main tasks:

- File Copy
  In this task, all the files on Android device are copied onto the SD card.
- Hashing
  This task ensures integrity of the copied files and allows discovering if some file got changed during the copy process.

The Acquisition Algorithm uses Android OS API's for performing various functions during the above process. This algorithm preserves the original directory structure, by copying files according to their original position recursively. The hashing ensures integrity check before and after copy of each file from Android device to the SD card data dump. The hashes are also written in Checksum.xml log file in home root directory which can be used for further verification. The Acquisition Algorithm invokes the hash function before and after copy of each file, ensuring verification of changes if any during the file copy.

CONCLUSION AND FUTURE WORK

Smartphones are becoming even more sophisticated and able. Both law enforcement and the private sector need to invest time and money into learning about new operating systems and developing new forensic methods [28]. Android OS is already the most popular OS on smartphones [21] and many more devices like tablets, televisions, vehicles, gaming devices, notebooks etc are already running on Android OS. However forensic analysis and security engineers have struggled as there is lack of knowledge and supported tools for investigating these devices [1]. Android Forensics is a quite young and immature discipline, even more when contextualized to the Mobile Forensics.

This paper outlined Efficient Generalized Forensics Framework for extraction and documentation of evidence from Android devices. This approach will ensure to acquire a complete and consistent snapshot of Android devices with through integrity verification using hashing algorithms. This study will be further used to do experimental analysis and relevant comparison with other commercial forensics tools available in market..

REFERENCES

[1] Andrew Hoog, "Android Forensics: Investigation, Analysis and Mobile Security for Google Android", Syngress Publication, 2011.

[2] Rizwan Ahmed, Dr. R. V. Dharaskar, "Android Forensics: Background, techniques and analysis tools", The proceedings of International Conference on Recent Advances In Information Technology (RAIT), 2012, pp. 265-269.

[3] IBM, "IBM X-Force Report Reveals Mobile Security Exploits to Double in 2011", http://www-03.ibm.com/press/us/en/pressrelease/35530.wss

[4] Christian Papathanasiou and Nicholas J. Percoco (2010),"This is not the droid you're looking for...", http://www.defcon.org/images/defcon-18/dc-18-presentations/Trustwave-Spiderlabs/DEFCON-18-Trustwave-Spiderlabs-Android-Rootkit-WP.pdf

[5] McAfee , "McAfee: Android malware surges 76%",
http://www.electronista.com/articles/11/08/23/mcafee.shows.android.facing.huge.spike.in.malware /

[6] Artem Russakovskii, "Massive Security Vulnerability In HTC Android Devices (EVO 3D, 4G, Thunderbolt, Others) Exposes Phone Numbers, GPS, SMS, Emails Addresses, Much More" http://www.androidpolice.com/2011/10/01/massive-security-vulnerability-in-htc-android-devices-evo-3d-4g-thunderbolt-others-exposes-phone-numbers-gps-sms-emails-addresses-much-more/

[7] More Android vulnerabilities exposed [Video Demonstration] : The Hacker News ~ http://thehackernews.com/2011/09/more-android-vulnerabilities-exposed.html

[8] Ryan Paul, "Android vulnerability reflects need for more timely updates", http://arstechnica.com/gadgets/news/2011/05/android-vulnerability-reflects-need-for-more-timely-updates.ars

[9] SWGDE and SWGIT Digital & Multimedia Evidence Glossary, SWGIT Digital & Multimedia Evidence Glossary Version: 2.3 (2009).

[10] Jansen, W., Ayers, R.: Guidelines on Cell Phone Forensics, NIST, SP 800-101 (2007).

[11] Rizwan Ahmed, Rajiv V. Dharaskar, "Mobile Forensics: an Overview, Tools, Future trends and Callenges form Indian Law perspective", Proceedings of International Conference on e-Governance (IIT, Delhi), 2008, pp. 312-323. http://www.iceg.net/2008/books/2/34_312-323.pdf

[12] Rizwan Ahmed, Rajiv V. Dharaskar, "Mobile Forensics: An Introduction from Indian Law Perspective", INFORMATION SYSTEMS, TECHNOLOGY AND MANAGEMENT, Communications in Computer and Information Science, 2009, Volume 31, Part 7, pp. 173-184. http://link.springer.com/chapter/10.1007%2F978-3-642-00405-6_21?LI=true#

[13] Rizwan Ahmed, Dr. R. V. Dharaskar, "MFL3G: An Open Source Mobile Forensics Library for Digital Analysis and Reporting of Mobile Devices for Collecting Digital Evidence, an Overview from Windows Mobile OS perspective", The proceedings of International Conference on Advanced Computing Technologies, 2008, pp. 245-256. http://www.gbv.de/dms/tib-ub-hannover/61808083x.pdf

[14] Zhang, Z.H., Luo, H.Y., Chen, L.X., Chen J.Y.: Digital home appliances industry trends, Ministry of Economic Affairs, R.O.C. (in Chinese)(2002)

[15] Ayers, R., Jansen, W., Moenner, L., Delaitre A.: Cell Phone Forensic Tools: An Overview and Analysis update, NISTIR 7387 (2007).

[16] Andy Rubin, "Where's my G-phone", http://googleblog.blogspot.com/2007/11/wheres-my-gphone.html (2007)

[17] Google Code, "ADC I Top 50 Gallery", http://code.google.com/android/adc/adc_gallery/

[18] Google, "Anrdroidology" , http://www.youtube.com/watch?v=QBGfUs9mQYY

[19] Paul, Ryan, "Dream(sheep++): A developer's introduction to Google Android". Ars Technica. Retrieved 2009-03-07.

[20] Android Open Source Project, http://source.android.com/index.html

[21] Canalys research release 2011/013, http://www.canalys.com/pr/2011/r2011013.html

[22] Alessandro Distefano, Gianluigi Mea, Francesco Pace, "Android anti-forensics", http://www.dfrws.org/2010/proceedings/2010-310.pdf

[23] YAFFS official website [Online]. Available: http://www.yaffs.net

[24] Kinam K. "Memory Technologies for Mobile era", Asian Solid-State Circuits Conference, 2005, pp. 7-11.

[25] An Introduction to NAND Flash [Online]. Available: http://www.commsdesign.com/showArticle.jhtml?articleID¼183700957; 2006.

[26] Android Security Architecture [Online]. Available: http://developer.android.com/guide/topics/security/security.html;2010.

[27] Nielsen, "More US Consumers Choosing Smartphones as Apple Closes the Gap on Android", http://blog.nielsen.com/nielsenwire/consumer/more-us-consumers-choosing-smartphones-as-apple-closes-the-gap-on-android/.

[28] Jeff Lessard and Gary C. Kessler, "Android Forensics:Simplifying Cell Phone Examinations", http://www.ssddfj.org/papers/SSDDFJ_V4_1_Lessard_Kessler.pdf

[29] Rizwan Ahmed, Dr. R. V. Dharaskar, "Mobile Forensics: the roadblocks ahead, proposed solutions using Protocol Filtering and SIM programming", International Journal Of Computer Science And Applications Vol. 2, No. 2. pp. 109-115.

[30] http://www.researchpublications.org/IJCSA/issue5/2009-IJCSA-5-5.pdf

[31] Rizwan Ahmed, Rajiv V Dharaskar, "Study of Mobile Botnets: An Analysis from the Perspective of Efficient Generalized Forensics Framework for Mobile Devices", IJCA Proceedings on National Conference on Innovative Paradigms in Engineering and Technology (NCIPET 2012) ncipet(15):5-8, March 2012. Published by Foundation of Computer Science, New York, USA. http://www.ijcaonline.org/proceedings/ncipet/number15/5302-1114