Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.25, NO.5, Oct. 2015 ISSN 2288-2715(Online)
http://dx.doi.org/10.13089/JKIISC.2015.25.5.1167

Firefox OS 포렌식 기법에 관한 연구

김 도 수,† 최 종 현, 이 상 진‡
고려대학교 정보보호대학원

The Study on Forensic Methodology of Firefox OS

Do-Su Kim,† Jong-hyun-Choi, Sang-jin Lee‡
Center for Information Security Technologies, Korea University

요 약

모바일 시장에서 스마트폰의 점유율이 기하급수적으로 높아짐에 따라 많은 제조사들은 자체적으로 모바일용 운영체제를 개발해왔다. Firefox OS는 Mozilla 재단이 개발한 스마트폰 및 태블릿용 오픈소스 운영체제이며 자바 스크립트를 사용하고 HTML5를 기반으로 동작한다. Firefox OS를 탑재한 스마트폰을 출시하는 제조사는 지속적으로 늘고 있는 추세이다. 하지만 기존의 Firefox OS 포렌식에 관한 연구는 단순히 추상적인 포렌식 프로세스 및 블록 크기에 따른 이미징 속도에 관한 연구만 진행되었기 때문에 조사관점에서 스마트폰에 존재하는 아티팩트들을 분석하기 힘들었다.

본 논문에서는 Firefox OS에서 사용자 데이터 훼손을 최소화하여 데이터를 수집하는 방법과 스마트폰에 남는 시스템 및 사용자 데이터를 분석한 결과를 바탕으로 포렌식 분석 프레임워크를 제안한다.

ABSTRACT

As the market share of smartphone exponentially increases in mobile market, a number of manufacturers have developed their own operating system. Firefox OS is an open source operating system for the smartphone and tablet which is being developed by the Mozilla Foundation. This OS is designed using JavaScript and operated based on HTML5. Even though the number of manufacturers which release the Firefox OS smartphone is consistently increasing, However it is difficult to analyze artifacts in a smartphone in terms of investigation since existing researches on Firefox OS focused on imaging velocity according to abstract forensic process and block size.

In this paper, we propose how to collect data in Firefox OS while minimizing data loss and forensic analysis framework based on analysis results on system and user data leaving in a smartphone.

Keywords: Forensics, Smartphone Forensics, Data Acquisition, Firefox OS, yaffs2, IndexedDB

I. 서 론

모바일 시장에서 스마트폰의 점유율이 기하급수적으로 높아짐에 따라 많은 제조사들은 자신들의 스마트폰 점유율을 높이기 위해 자체적으로 모바일용 운영체제를 개발해왔다. Firefox OS는 Mozilla 재단이 개발한 스마트폰 및 태블릿용 오픈소스 운영체제이다. 중저가 시장을 겨냥하고 있으며 저사양에서도 동작하도록 설계되었다. Firefox OS와 같은 Web OS는 리눅스 커널을 기반으로 하며, 특정 API에 종속적인 기존 안드로이드 및 iOS 애플리케이션에 비해 HTML5, Java Script, CSS 등의 웹 표준 기술로 빠르고 쉽게 애플리케이션을 제작할 수 있다. 다양한 모바일 플랫폼이 등장함에 따라 플랫폼에 의

접수일(2015년 7월 14일), 수정일(1차: 2015년 8월 13일, 2차: 2015년 8월 20일), 게재확정일(2015년 9월 5일)
† 주저자, kimdosu@korea.ac.kr
‡ 교신저자, sangjin@korea.ac.kr(Corresponding author)

존적이지 않은 HTML5를 이용한 웹 애플리케이션을 사용하는 비율이 점차 증가하고 있다[1]. Web 기반 OS들이 지속적으로 출시되고 있으며 Firefox OS를 탑재한 스마트폰을 출시하는 제조사 또한 늘고 있는 추세이다[2]. 또한 Firefox OS 커스텀 롬을 안드로이드 스마트폰에 플래싱하여 사용할 수 있기 때문에 분석이 미비한 점을 사용자가 악용할 수 있다. 실제로 많은 범죄에 스마트폰이 사용되고 있으며 디지털 데이터가 디지털 증거로 사용되기 위해서는 데이터 무결성과 원본성이 입증할 수 있어야 하지만 Firefox OS에 존재하는 데이터에 관한 연구는 미비한 실정이다. 따라서 조사과정에서 Firefox OS 포렌식의 필요성은 점차 증가할 것이며 본 논문에서 제안하는 Firefox OS 포렌식 기법의 활용도 또한 증가할 것이다.

본 논문의 연구결과는 기존 연구에 진행되지 않은 데이터 파티션 훼손을 최소화하여 수집하는 방법에 대한 실험과 실제 데이터 분석을 토대로 획득 가능한 데이터 분류와 분석 방법을 연구하고 실제 조사 환경에서 활용할 수 있는 포렌식 분석 프로세스를 제안하는 방향으로 연구를 진행하였다. 2장에서는 Firefox OS와 모바일 포렌식과 관련된 기존 연구에 대해 설명한다. 3장에서는 Firefox OS 시스템, 사용자 데이터 수집 방법과 분석 방법에 대해 프레임워크를 제안한다. 4장에서는 3장에서 제안한 프레임워크를 바탕으로 Firefox OS 포렌식 프로세스를 제안하고 5장에서 결론을 맺는다.

II. 관련 연구

2.1 포렌식 관점의 루팅 기법

디지털 포렌식 관점에서 디바이스 내 사용자 데이터 추출 시 훼손을 최소화 하는 것은 매우 중요한 일이다. Yunho Lee는 안드로이드 기반의 스마트폰 내부 정보 추출을 위한 루팅 시 사용자 데이터 훼손을 최소화 하는 방법에 대해 설명하고 있다[3]. 논리적인 안드로이드 루팅 방법에는 루팅된 롬 이미지 플래싱과 앱을 이용한 루팅이 있으며 안드로이드에서 사용자 데이터의 훼손을 최소화 할 수 있는 루팅 방법은 롬 플래싱 이라고 제시하고 있다. Firefox OS 또한 필요시 사용자 데이터 훼손을 최소화하여 루트 권한을 획득해야 한다.

2.1 수집 기법

Firefox OS 데이터 수집의 경우 안드로이드와 같이 커스텀 리눅스 커널을 사용하므로 동일한 방법을 통해 이미징할 수 있으며 Timothy Vidas는 안드로이드 환경에서 Yaffs2 파일시스템 데이터 수집 방법에 대해 설명하고 있다[4].

Mohd Najwadi Yusoff는 Firefox OS에 대한 포렌식 준비, 데이터 획득 프로세스에 대해 추상적으로 설명하고 있으며 블록 크기 별 이미징 속도를 테스트 하였다. 하지만 실제 분석에 활용하기 위해서는 Firefox OS에서 수집할 수 있는 시스템, 사용자 데이터에 대한 분류나 어떠한 데이터를 수집하여 분석에 활용해야 하는지에 대해 제시되어야 하지만 이에 대한 연구는 진행되지 않았다.[5,6,7]

2.2 분석 기법

Mohd Najwadi Yusoff는 Firefox OS 분석 프로세스를 제시하고 있다. 하지만 스마트폰 내부 증거 데이터 확보를 위한 구체적인 시스템 및 사용자 데이터 분석 방`법에 대한 설명은 하지 않았다[5,6].

Andrew Hoog는 안드로이드 기반 스마트폰에 대한 분석 방법과 애플리케이션 별 아티팩트들을 나열하고 있다[8]. 안드로이드와 같이 기본 애플리케이션에 대한 분석 기법은 Firefox OS에서 적용하여 활용할 수 있다.

위와 같이 기존에 연구된 내용들은 안드로이드의 루팅, 수집, 분석 기법과 Firefox OS의 추상적인 포렌식 프로세스에 대해 연구되었으나, 실제 분석 환경에서 적용할 수 있는 Firefox OS의 구체적인 포렌식 아티팩트 및 데이터 분석에 대한 언급은 하지 않았다.

III. Firefox OS 기반 스마트폰 포렌식 Framework

3.1 데이터 수집

Firefox OS의 경우 데이터 수집을 위해 필요시 사용자 데이터의 훼손을 최소화 하는 방법으로 루트 권한을 획득해야 하는데 가능한 루팅 방법으로는 취약점 익스플로잇과 루팅된 롬 이미지 플래싱 방법이 있다[9,10]. Firefox OS에서 사용자 데이터 훼손

을 최소화하는 루팅 방법에 관한 실험을 위해 Ext4 파일시스템을 사용하는 Nexus S 레퍼런스 스마트폰에 Firefox OS를 설치하여 루팅 기법에 따른 data 파티션의 변화를 블록 단위로 비교하였다.

Table.1.은 안드로이드 스마트폰인 Nexus S에 Firefox OS를 설치한 후 롬 플래싱 기법을 5번 반복하여 수행한 후 data 파티션의 해시값을 비교한 표이다.

롬 플래싱을 5번 반복 실험한 결과 data 파티션의 해시값은 동일하였다. 롬 플래싱 후 익스플로잇을 통한 루팅이 data 파티션에 미치는 영향을 확인하기 위해 롬 플래싱 되어 있는 스마트폰에 익스플로잇을 진행하였다. 그 결과 data 파티션의 해시값이 변화함을 알 수 있었다.

Table 2.는 롬 플래싱을 한 Firefox OS에 익스플로잇을 한 이미지의 훼손 정도를 바이트 단위로 비교한 표이다.

블록 단위로 비교한 결과 취약점 익스플로잇을 사용할 경우 블록들이 상당수 변화하였다. 취약점 익스플로잇의 경우 데이터 영역에 익스플로잇 코드를 넣어야 하기 때문에 데이터 영역이 훼손되지만 롬 플래싱의 경우 fastboot 프로토콜을 사용하여 시스템 파티션만 마운트되고 데이터 파티션은 마운트가 되지 않기 때문에 포렌식 관점에서 데이터의 훼손 없이 루팅할 수 있다[3].

추가적으로 시스템 로그들을 기록하는 cache 파티션을 5회 반복 실험한 결과 롬 플래싱 기법과 익스플로잇 기법 모두 해시값의 변화는 없었다.

따라서 Firefox OS의 사용자 데이터 훼손을 최소화 하는 루팅을 진행할 경우 롬 플래싱 기법을 사용하여 루팅해야 한다.

루트 권한을 획득한 후 Mohd Najwadi Yusoff는 ext4 기반의 Firefox OS에서 블록 장치 전용 파일시스템의 논리적 데이터 이미지를 생성해주는 DD 바이너리를 이용하여 파티션을 이미징하는 방법을 제시했다[5].

Table 1. Compare data Partition MD5 Hash Value after Rom Flashing

Count	Hash(MD5)
1	47b19d21798a8850199ed317d25f3a5d
2	47b19d21798a8850199ed317d25f3a5d
3	47b19d21798a8850199ed317d25f3a5d
4	47b19d21798a8850199ed317d25f3a5d
5	47b19d21798a8850199ed317d25f3a5d

Table 2. Ext4 Byte Comparision

	Equal	Difference
Exent Blocks	266,240	-
Volume Bitmap	32,764	4
Journal Area	16,229,041	548.175
Inode Bitmap	32,765	3
Group Descriptors	20,472	8
Inode Table	16,775,373	1,843
Block Descriptors	20,468	12
Allocated Cluster	94,356,938	418,413
Unallocated Clusters	978,172,937	1,684,471
Total	1,105,906,998	2,652,929

하지만 저사양 스마트폰 지원에 초점을 둔 Firefox OS는 Yaffs2 파일시스템도 사용하기 때문에 Yaffs2 기반의 파티션 수집 기법에 대한 연구도 필요하다. NAND Flash 기반의 Yaffs2를 사용하는 스마트폰에서는 nanddump 도구를 이용하여 MTD(Memory Technology Device)에서 제공하는 인터페이스를 통해 NAND flash의 스페어 영역까지 포함하여 이미징을 해야 한다[11].

실험을 위해 Yaffs2 파일시스템을 사용하는 Firefox OS 스마트폰인 ZTE Open을 사용하였다[12].

adb에서 root 권한으로 접근한 후 mount 명령어를 통해 data 파티션의 마운트 정보를 확인한다. 그 후 nanddump를 이용하여 data 파티션을 sdcard로 이미징하는 명령어는 다음과 같다.

```
./nanddump -a /dev/mtd/mtd6 -f /sdcard/userdata.dd -o -bb=padbad
```

이미징 후 분석할 데이터 선별 과정이 필요하다. 시스템 데이터와 애플리케이션 데이터를 분류한 후 각각의 아티팩트 경로를 정리한 표는 Table 3.과 Table 4.와 같다.

Table 3. System Log List

Name	Path	Mount Partition
System log	/logs/kernel/log_kernel.txt	cache
	/logs/logcat/logcat_ps.txt	
	/logs/resetlog/smem_log_power_events.txt	
Recovery log	/recovery/log	
Bluetooth	/misc/bluetoothd/[Mac Address]/name	data
	/misc/bluetoothd/[Mac Address]/lastseen	
Wifi	/misc/wifi/wpa_supplicant.conf	

Table 4. Application Data List

App Name	DB Path
Manifest	/local/webapps/[appID]
Firefox cookie	/b2g/Mozilla/[Random Character].default/cookies.sqlite
Firefox History	/local/indexedDB/[AppID]+app+++browser.gaiamobile.org/[Random Number]brreosw.sqlite
E-mail	/local/indexedDB/[AppID]+f+app+++email.gaiamobile.org/[Random Number]bl2iga-me.sqlite
Contacts	/local/indexedDB/chrome/[Random Number]csotncta.sqlite
Calendar	/local/indexedDB/[AppID]+app+++calendar.gaiamobile.org/[Random Number]br2agd-nceal.sqlite
Notes	/local/indexedDB/[AppID]+app+++note.gaiamobile.org/[Random Number]EsVeMtEo_N.sqlite
SMS	/local/indexedDB/chrome/[Random Number]ssm.sqlite
Call History	/local/indexedDB/[AppID]+f+app+++communications.gaiamobile.org/[Random Number]dsitanleecreR.sqlite

3.1.1 시스템 데이터

분석 대상 시스템 데이터에는 커널 로그, 프로세스 로그, 리커버리 로그, 블루투스, 와이파이 연결정보가 있다.

3.1.2 애플리케이션 데이터

분석 대상 Firefox OS 기본 애플리케이션에는 웹 브라우저, 문자, 통화내역, 캘린더, 이메일, 연락처, 노트가 있다. 각각의 애플리케이션들은 고유한 경로에 DB를 저장하고 있으며 /local/webapps/webapps.json 파일에 해당 디바이스에 설치된 애플리케이션들의 정보가 저장되어 있다.

3.2 데이터 분석

3.2.1 시스템 데이터

cache 파티션 분석을 통해 kernel 로그, logcat 로그, reset 로그, Recovery 로그를 획득할 수 있다. smem_log_power_events 로그 파일에 기록되는 power event time과 log_kernel.txt 파일에 기록되는 커널 행위와 동작 시각을 확인하여 OS 구동 시각을 알아낼 수 있다.

OS 동작 시각을 확인한 후 logcat_ps.txt 파일에 저장되는 프로세스 정보를 확인하면 사용자가 프로세스를 언제 실행했는지 확인할 수 있다.

recovery/log는 recovery 시각과 데이터 와이핑 시각, 파일시스템 정보를 저장하며 Fig.1.과 같다.

데이터 와이핑 테스트 결과 Firefox OS는 데이터 영역을 제로라이징하는 것을 확인할 수 있었다.

따라서 시스템 로그를 분석 시 recovery log에 데이터 와이핑 기록이 있다면 해당 디바이스의 데이

```
Starting recovery on Sun Jan  6 00:05:39 1980
framebuffer: fd 4 (320 x 480)
recovery filesystem table
=========================
  0 /tmp ramdisk (null) (null) 0
  1 /boot mtd boot (null) 0
  2 /amss mtd amss (null) 0
  3 /appsbl mtd appsbl (null) 0
  4 /mibib mtd mibib (null) 0
  5 /qcsbl mtd qcsbl (null) 0
  6 /oemsbl1 mtd oemsbl1 (null) 0
  7 /oemsbl2 mtd oemsbl2 (null) 0
  8 /splash mtd splash (null) 0
  9 /cache yaffs2 cache (null) 0
  10 /data yaffs2 userdata (null) 0
  11 /misc mtd misc (null) 0
  12 /recovery mtd recovery (null) 0
  13 /sdcard vfat /dev/block/mmcblk0p1 /dev/block/mmcblk0 0
  14 /system yaffs2 system (null) 0

I:Got arguments from /cache/recovery/command
mtd: successfully wrote block at 0
I:Set boot command "boot-recovery"
Command: "/sbin/recovery" "--wipe_data"
```

Fig. 1. Recovery log data

터를 사용자가 의도적으로 삭제한 행위이기 때문에 분석시 참고해야 한다.

data 파티션에 는 Bluetooth, Wifi 연결 정보가 저장된다. 블루투스 연결 정보를 담고있는 name 파일은 기기이름, 연결 대상 기기의 MAC 주소, 최근연결시간 등을 저장하며 와이파이 연결 정보를 담고 있는 wpa_supplicant.conf 파일은 기기이름, psk(패스워드), 암호화 방식을 저장한다.

따라서 시스템 데이터 분석을 통해 획득할 수 있는 정보는 Table 5.와 같다.

Table 5. Obtainable System Artifacts

Mount Partition	Artifacts	Data
Cache	Kernel Log	Kernel Action Time, Kernel Action
	Process Log	PID, UID, Thread Name, CPU Utilization Rate
	Power Event Log	Power Event Time
	Recovery Log	Recovery Time, Wiping Time, Partition Table
Data	Bluetooth	Device Name, MAC Address, Lately Access Time
	Wifi	Device Name, Password

3.2.2 애플리케이션 데이터

Firefox OS의 애플리케이션 데이터에는 App 설치 정보, cookie, history, 각각의 App 데이터 들이 있다. Web 기반의 스마트폰 특성상 저사양 하드웨어에서 동작해야 하며 네트워크 데이터 송수신 속도에 따라 성능의 상당부분이 결정된다. 따라서 데이터 압축이 필요한데 Firefox OS는 Google에서 내부 파일시스템에 사용하다가 2011년도에 오픈소스로 공개한 Snappy 압축 알고리즘을 사용하여 history와 App 데이터를 압축하여 저장한다[13].

따라서 압축된 데이터의 경우 분석을 위해 압축을 해제하여 원본 데이터를 획득하는 과정이 필요하다.

3.2.2.1 압축되지 않은 데이터

압축되지 않은 애플리케이션 데이터에는 manifest 파일과 Firefox cookie DB가 있다. Webapp의 경우 /local/webapps/[appID]의 경로에 각각의 애플리케이션 정보를 나타내는 manifest.webapp 파일을 저장하고 있다. 해당 파일 분석을 통해 애플리케이션 제공 사이트, 설치시간, 로컬ID 등의 정보를 획득할 수 있다.

Fig.2.는 Firefox 쿠키 파일인 cookies.sqlite의 내용이다. Firefox OS는 사용자 애플리케이션들이 접근하는 URL을 AppID와 접근시각 등과 함께 cookies.sqlite 파일에 저장한다.

안드로이드의 경우 웹앱을 통해 생성된 쿠키와 브라우저를 통해 생성된 쿠키를 각각의 경로에 따로 저장하지만 Firefox OS는 하나의 DB 파일에 통합하여 저장한다.

따라서 cookies.sqlite 파일 분석을 통해 각각의 애플리케이션 또는 브라우저로 접근한 도메인과 접근시간에 대한 정보를 획득할 수 있다.

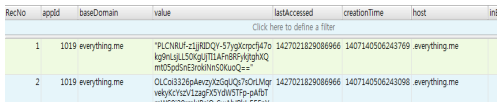

Fig. 2. Firefox Cookies DB

3.2.2.2 압축된 데이터

Fig.3.은 note 애플리케이션의 IndexedDB 테이블 구조를 나타낸다. IndexedDB는 사용자의 웹 브라우저 정보가 통합된 클라이언트 사이드 DB이다. 관계형 DB의 경우 저장되는 데이터의 스키마 유연성이 떨어질 수 있고 SQL이라는 독립 언어를 기반으로 하기 때문에 브라우저간 표준화 및 호환성에 문제가 될 수 있다. 반면 IndexedDB는 트랜잭셔널 DB 모델을 따르며 SQL 언어와는 무관하다. Firefox OS는 단순한 저장구조(Key-Value Storage)를 갖고 있는 IndexedDB를 sqlite 포맷으로 저장한다. 애플리케이션이 온라인일 경우 IndexedDB에 데이터를 저장하며 인터넷 연결이 종료되었을 경우 IndexedDB의 데이터를 가져와서 오프라인에서 동작한다[14].

Fig.4.는 Calendar 어플리케이션의 DB인 br2agd-nceal.sqlite의 내용이다. 해당 DB에는

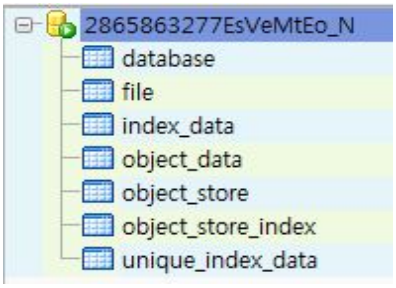

Fig. 3. IndexedDB Table Structure

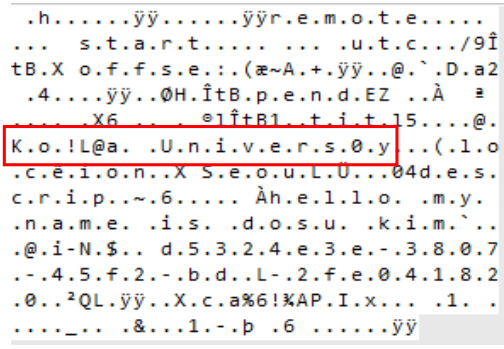

Fig. 4. Compressed Calendar DB Data

캘린더에 추가한 일정 정보가 BLOB 형태로 압축되어 저장되어 있다.

Firefox OS 소스코드를 분석한 후 압축 해제 코드를 작성하여 Calendar DB 데이터 압축을 해제한 그림은 Fig.5.와 같다.

각 항목별 데이터 구분자는 0xFFFF이며 그 뒤로 데이터가 저장된다. 기존에 압축되어 정확히 확인할 수 없었던 항목 및 데이터를 압축해제를 통해 원본을 획득할 수 있다.

압축해제를 통해 획득할 수 있는 애플리케이션별

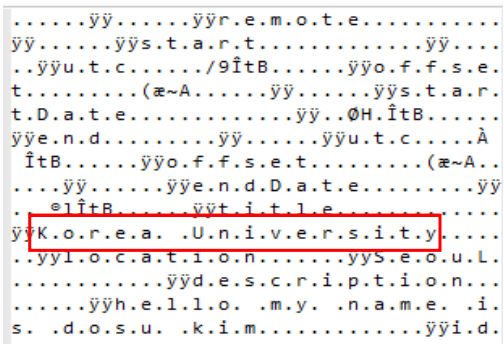

Fig. 5. Decompressed Calendar DB Data

Table 6. Obtainable App Artifacts

App Name	Artifacts	Compression
App Manifest	App Install Info	Non Compressed
Firefox cookie	URL	
Firefox History	URL	Compressed
E-mail	Mail ID, Password	
Contacts	Contact, Name, Group	
Calendar	Schedule, Work	
Notes	Memo, Memo Time	
SMS	SMS Content, SMS Time	
Call History	Phone Number, Sender, Receiver, Call Time	

아티팩트는 Table 6.과 같다.

IV. 제안하는 Firefox OS 포렌식 프로세스

Mohd Najwadi Yusoff가 제안한 포렌식 프로세스는 모든 스마트폰 OS에서 사용할 수 있는 일반적인 스마트폰 포렌식 프로세스이다. 하지만 기존 연구는 Firefox OS에서 수집할 수 있는 데이터에 대한 분류나 분석 방법에 대한 연구는 진행되지 않았다.

따라서 분석 환경에서 활용하기엔 상당히 추상적인 프로세스이기 때문에 본 논문에서 분석한 결과를 토대로 실제 데이터 분석에 초점을 맞춰 제안하는 Firefox OS 포렌식 프로세스는 Fig.6.와 같다.

앞서 실험한 data 파티션 훼손 결과에 따라 루팅된 롬을 보유하고 있을 경우 롬 플래싱을 통해 루트 권한을 획득하고 롬이 없을 경우 취약점 익스플로잇을 통해 루트 권한을 획득한다.

Firefox OS는 Yaffs2와 Ext4 둘다 지원하기 때문에 Yaffs2일 경우 Nanddump 바이너리를 사용하고 ext4일 경우 DD 바이너리를 사용하여 분석하고자 하는 파티션을 이미징한 후 시스템 데이터와 애플리케이션 데이터를 분류한 후 각각의 경로에서 분석할 데이터들을 수집한다. 애플리케이션 데이터의 경우 IndexedDB BLOB 데이터들이 snappy 알

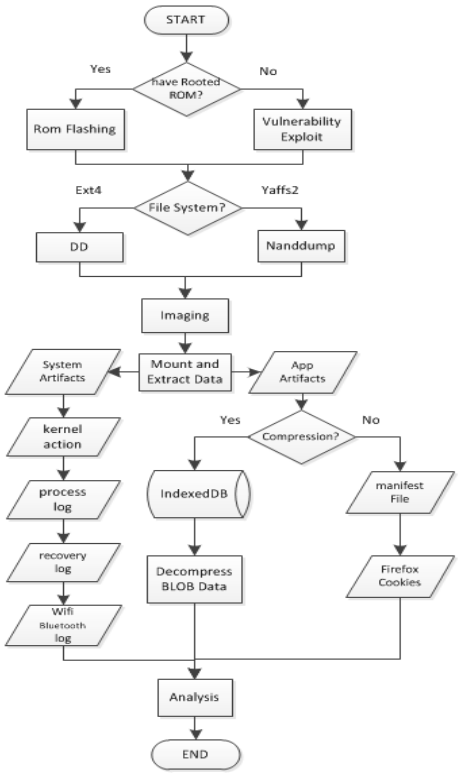

Fig. 6. Firefox OS Forensic Process

고리즘으로 압축되어 있으므로 압축 해제 과정을 통해 원본 데이터를 획득하고 분석한다.

따라서 3장에서 분류, 분석한 데이터들을 토대로 기존에 Mohd Najwadi Yusoff가 제안한 Ext4 파일시스템을 사용하는 Firefox OS 포렌식 프로세스 뿐만 아니라 Yaffs2 파일시스템에 대한 수집 방법과 데이터 분류, 획득한 데이터의 처리 방법에 관한 프로세스를 제안하였다.

V. 결 론

본 논문에서는 기존 연구에서 더 나아가 Firefox OS 스마트폰에 남는 시스템, 사용자 데이터들의 분석 방법을 제시하고 실제 아티팩트들을 분석하였다.

또한 Snappy 알고리즘을 통해 압축되어 있던 사용자 데이터들을 압축 해제하여 원본 데이터를 획득하였고 이를 통해 Firefox OS에 남는 아티팩트들을 분석할 수 있음을 보였다.

따라서 본 논문에서 제시하는 포렌식 분석 프레임워크를 활용함으로써 사용자 데이터 훼손 없이 기존에 분석할 수 없었던 사용자 데이터를 정상적으로 분석할 수 있다.

기존에 높은 점유율을 가지고 있는 안드로이드에 비해 많은 분석이 이루어지지 않았기 때문에 Firefox OS의 경우 잔존하는 데이터를 조사하기 힘들다. IndexedDB를 사용하는 애플리케이션의 경우 압축 해제 과정을 통해 원본 데이터를 획득한 후 분석하면 포렌식적으로 유용한 정보를 많이 획득할 수 있기 때문에 시스템 로그, GPS 정보, 사용자 애플리케이션 데이터에 관한 추가적인 연구가 필요하다. 또한 오픈소스 OS인 Firefox OS는 다양한 버전이 존재한다. 스마트폰 이외에도 태블릿, 스마트 TV 등에 사용되고 있는 만큼 새로운 버전의 업데이트 또는 제품이 출시되면 지속적인 관심을 가지고 확인해야 한다.

References

[1] http://www.developereconomics.com/cross-platform-apps-qt-vs-html5.

[2] https://www.strategyanalytics.com/strategy-analytics/blogs/devices/smartphones/smart-phones/2012/09/27/firefox-os-to-capture-1-percent-share-of-global-smartphone-market-in-2013#.VWQFgU0w-Ag.

[3] Yunho Lee, "A Method of Internal Information Acquisition of Smartphones," Journal of The Korea Institute of Information Security & Cryptology, vol. 23, no. 6, pp. 1060-1062, Dec, 2013.

[4] Timothy Vidas, "Toward a general collection methodology for Android devices," Digital Investigation 8, S14-S24, 2011.

[5] Mohd Najwadi Yusoff, "Advances of Mobile Forensic Procedures in Firefox OS," International Journal of Cyber-Security and Digital Forensics(IJCSDF), 3(3), pp. 149-151, 2014.

[6] Mohd Najwadi Yusoff, "An Approach for Forensic Investigation in Firefox OS," Cyber Security, Cyber warfare and

Digital Forensic(CyberSec), 2014 Third International Conference on IEEE , pp. 22-26, May. 2014.
〔7〕 Mohd Najwadi Yusoff, "Performance Measurement for Mobile Forensic Data Acquisition in Firefox OS," International Journal of Cyber-Security and Digital Forensics(IJCSDF), 2014.
〔8〕 Android forensics: investigation, analysis and mobile security for Google Android.
〔9〕 http://pof.eslack.org/2013/07/05/zte-open-firefoxos-phone-root-and-first-impressions.
〔10〕 https://developer.mozilla.org/en-US/Firefox_OS/Building_and_installing_Firefox_OS.
〔11〕 Darren Quick, "FORENSIC ANALYSIS OF THE ANDROID FILE SYSTEM YAFFS2", Australian Digital Forensics Conference, 5th-7th, Dec. 2011.
〔12〕 http://www.gsmarena.com/zte_open-5320.php.
〔13〕 snappy, "https://code.google.com/p/snappy", 2011
〔14〕 https://developer.mozilla.org/ko/docs/IndexedDB/Basic_Concepts_Behind_IndexedDB.

〈저자소개〉

김 도 수 (Do-Su Kim) 정회원
2014년 2월: 동명대학교 정보통신공학과 졸업
2014년 3월~현재: 고려대학교 정보보호대학원 석사과정
〈관심분야〉 정보보호, 디지털 포렌식

최 종 현 (Jong-Hyun Choi) 정회원
2012년 2월: 경희대학교 전자정보대학 컴퓨터공학과 졸업
2014년 8월: 고려대학교 정보보호대학원 석사
2014년 9월~현재: 고려대학교 정보보호대학원 박사과정
〈관심분야〉 정보보호, 디지털 포렌식

이 상 진 (Sang-Jin Lee) 종신회원
1987년 2월: 고려대학교 수학과 졸업
1989년 2월: 고려대학교 수학과 이학석사
1994년 8월: 고려대학교 수학과 이학박사
1989년 10월~1999년 2월: ETRI 선임 연구원
1999년 3월~2001년 8월: 고려대학교 자연과학대학 조교수
2001년 9월~현재: 고려대학교 정보보호대학원 교수
〈관심분야〉 디지털 포렌식, 정보은닉이론, 대칭키 암호