Chapter 4

FORENSIC EVALUATION OF AN AMAZON FIRE TV STICK

Logan Morrison, Huw Read, Konstantinos Xynos and Iain Sutherland

Abstract This chapter presents the results of a forensic acquisition and analysis of an Amazon Fire TV Stick, a popular streaming media device. Although the primary functions of the Fire TV Stick are streaming videos and playing non-intensive video games, it is a reasonably powerful device that runs an Android operating system. This chapter explores the additional capabilities being developed for Fire TV Sticks in the hacker/enthusiast community and considers the implications that alterations to the devices could have with regard to digital forensics. An empirical assessment is conducted to identify the potential for misuse of Fire TV Sticks and to provide guidance to forensic investigators who analyze these devices.

Keywords: Embedded systems, Fire TV Stick, forensic evaluation

1. Introduction

One aspect of digital convergence is the ability of users to replace wired entertainment systems such as cable television with wireless media streaming services [3]. Streaming service providers have responded to this demand by introducing devices like the Fire TV Stick. Such a device brings streaming services to a user's television set by merely attaching a USB device to the set without any other connections. Over time, these devices have become very popular – current estimates indicate that more than 50% of U.S. homes have a television set connected to the Internet via one of these devices. It is also estimated that global shipments will increase from 240 million devices in 2016 to 382 million devices by 2021 [7]. The increased use of these devices has caused streaming media forensics to become its own specialty area.

G. Peterson and S. Shenoi (Eds.): Advances in Digital Forensics XIII, IFIP AICT 511, pp. 63–79, 2017.
DOI: 10.1007/978-3-319-67208-3_4

During the first quarter of 2015, it was reported that there were 4.5 million Fire TV devices in use [8] and the number of these devices has surely grown since then. This figure incorporates Fire TV Sticks into its calculation. Meanwhile, there is little information about the data stored on a Fire TV Stick and how to acquire an image of the data in a forensically-sound manner. This chapter discusses the information stored on a Fire TV Stick that may be of interest to forensic investigators. Also, it presents guidance on conducting a forensic evaluation of a Fire TV Stick.

2. Related Work

Streaming media devices present unique challenges when it comes to accessing data, since many of them may require hardware modifications in order to access data. Some research related to the forensic acquisition and analysis of data from similar streaming media devices has been attempted, but little, if any, research has focused on the Amazon Fire TV Stick. It is possible to understand the challenges that may be experienced with regard to potential acquisition methods by reviewing work on similar media streaming devices and other small-scale devices.

2.1 Chromecast

Researchers have analyzed the files contained in a crash report generated by a Google Chromecast device [9]. However, this involves crashing the device, which causes major changes to the device; it is, therefore, less than ideal in a digital forensic investigation. The primary challenges are that the universal asynchronous receiver/transmitter (UART) connection provides minimal data and that the flash chip is encrypted with a unique (per device) key. These challenges make data acquisition and analysis very difficult. Nevertheless, the analysis of the crash report, which is in the form of a ZIP file, provides information about the layout of the NAND chip, useful timestamps and data pertaining to streamed videos.

2.2 Measy A2W Miracast

An analysis of the Measy A2W Miracast device [9] is more interesting because of the larger number of acquisition possibilities and the amount of useful data that can be recovered. Hardware experiments have accessed the UART interface by physically connecting to several pins on the main circuit board. This enables Hexdump to be used to extract a memory dump. However, employing the UART interface can change the device memory and, therefore, may not be forensically sound.

Experiments with the `curl` binary in the device firmware revealed that files can be posted to a Wi-Fi enabled server, but this is inconsistent and unreliable. An experiment that imaged the NAND flash chip revealed that the chip could be imaged effectively and in a forensically-sound manner using a toolkit with a write blocker [9]. Experiments using a Netcat listener to acquire files over Wi-Fi were also conducted.

Researchers were also able to recover MAC addresses, links, image files, URLs, firmware data, timestamps regarding device usage, WPA2 passwords and SSIDs using toolkits and techniques such as file carving [9]. This work is of interest because it discusses: (i) multiple methods for acquiring data from a streaming media device; (ii) a forensically-sound data acquisition technique, (iii) challenges that can arise when working with the devices; and (iv) potential methods for overcoming the challenges. However, some of the methods present risks that may keep them from being used in digital forensic investigations, including the possibility of permanently disabling (i.e., "bricking") the devices [2].

2.3 Amazon Kindle Fire HD

Research by Iqbal et al. [6] is of particular interest due to the similarities between the Kindle Fire HD and Fire TV Stick. Both devices run Amazon's Fire operating system and have an EXT4 file system [4]. Therefore, the experimental results for the Kindle Fire HD could be useful for the Fire TV Stick as well.

An experiment conducted by Iqbal et al. used a modified USB cable and a QEMU automated root exploit to gain root access. Next, the Android debug bridge (ADB) was used to image the userdata partition. The analysis of the userdata partition revealed that app data, user data, photographs, browsing data, audio data, cloudsync status and other useful data could be recovered.

The research on the Kindle Fire suggests an initial approach should focus on the userdata partition of a Fire TV Stick. It also demonstrates the challenges in achieving – and the importance of having – root access to the device in order to access the partition. However, this method requires the USB debugging function to be enabled on the device, which poses problems when this feature is disabled. Note that enabling USB debugging may result in the modification of the device and, thus, affect the forensic soundness of the recovered information.

3. Proposed Forensic Methodology

Forensic soundness is an extremely important characteristic of an evidence extraction methodology. This is accomplished by limiting, if not eliminating, the changes made to the evidentiary device before and/or during data extraction. Thus, all the experiments conducted on the Fire TV Stick in this research have paid special attention to this requirement.

A literature survey and an exploration of the Fire TV Stick functionality were performed to identify potential methods and areas of interest on the device. The research on Amazon Kindle Fire HD forensics by Iqbal et al. [6] was extremely useful from this perspective. The Fire TV Stick research involved reviewing its functionality as well as powering the device and going through the various menus to ascertain the kinds of artifacts that may reside on the device. The focus was to identify commonly-used functionalities, applications providing the functionalities and locations where artifacts related to user actions reside on the device. It was identified empirically that the userdata partition would be the most likely location to find artifacts of interest. Specifically, the following features of interest to a typical user and the applications associated with these features were identified:

- Video streaming through Netflix, YouTube and free Amazon content.
- Music streaming through Spotify and Pandora.
- Gaming through Amazon's App Store.
- Uploading/viewing photographs through Amazon's Cloud Drive.
- App downloading through Amazon's App Store.
- Sideloading of Android apps through the Android debug bridge.

3.1 Experimental Methodology

Table 1 summarizes the experimental methodology. The methodology was developed by exploring the device functionality, device state (on/off) and various physical, logical and manual acquisition options. The goal was to determine if it was possible to retrieve data generated by the various features and applications. This involved simulating typical user behavior using the available features and applications in order to introduce data and discover if it could be retrieved for subsequent analysis. Towards the end of the experiment, a new software update became available for the Fire TV Stick (version 5.0.5.1). This update caused problems with some of the acquisition methods. Due to time constraints, the update and its effects could not be explored fully.

Table 1. Amazon Fire TV Stick experimental methodology.

Method	**Description**
Evaluate Fire TV Stick Condition	Examine a new out-of-the-box device with/without USB debugging enabled, with/without root enabled, before/after user interactions.
Select Method	Select physical method (ADB raw device imaging with root), logical method (file copying with custom Python script and ADB) or manual method (visual inspection of menus, etc.).
Activate Video Capture	Record the time, create a record of actions.
Power on the Device	Record time, note boot sequence of device, default menus, etc.
Empirically Assess Features	Systematically traverse through the identified features and applications, interact with them, record content consumed/created, observations and times for future retrieval.
Power off the Device	Record the time, turn off the Fire TV Stick and video capture device.
Attempt Data Recovery	Create an image of the Fire TV Stick using a physical, logical or manual method. Record the success or failure of the method and any pertinent image-specific data.
Investigate Images	Use forensic tools (FTK Imager v.3.4.0.1, AccessData Labs v.5.1) to retrieve data.

3.2 Sample Data

The Fire TV Stick is designed to be registered to a specific Amazon account in order to access Amazon content. An Amazon account was created for the device under the name "Fire Stick." A gaming profile was also created locally on the device under a pseudonym. An author's accounts for Netflix, Pandora and Spotify were used to test the video and music streaming features.

To assess the video streaming features, Netflix and YouTube applications were installed and accessed. The Netflix application was then used to stream the first ten minutes of several sample movies and television shows. The YouTube application was used to stream several sample videos. Finally, Amazon's free streaming video content was used to stream sample videos.

The assessment of the music streaming feature involved the installation of the Spotify and Pandora apps. Spotify was used to stream several sample music tracks. The Pandora app was used to stream music from a radio station.

The gaming feature was assessed by first setting up a local gaming profile on the Fire TV Stick. This profile was created and named automatically without direct action by the user. It appeared after the first game was downloaded and launched on the system. Amazon's App Store was then used to download two sample games, *Flappy Birds Family* and *Crossy Road*. The apps were then launched individually and two rounds of each game were played.

In order to assess the photograph uploading and viewing feature via Amazon's Cloud Drive, a free trial for the Cloud Drive was obtained using the Fire Stick Amazon account and an email address. Photos were then uploaded using the Fire Stick account, Cloud Drive website and a desktop personal computer. After the photographs were uploaded, they were viewed via the Fire TV Stick's photo tab. The photo tab was then used to view the Test and Favorites albums and to add the photographs to the albums.

Amazon's App Store and the app downloading feature were assessed by downloading additional programs. NBC and HBO Go apps were downloaded from the App Store to assess this feature. The idea was to see the kind of information that could be recovered about apps that were downloaded but never used.

The sideloading of Android apps from sources other than Amazon's App Store was assessed using the Android debug bridge and ES File Explorer. ES File Explorer was used to download Kodi, formerly known as the Xbox Media Center, directly from its download page. This was done by creating a Favorites tab in Kodi to navigate to a web page and using the remote to navigate the website. Kodi was then launched to ensure that it worked properly. The `install` command of the Android debug bridge can be used from a workstation to obtain downloaded Android APK files and install them on the Fire TV Stick. This method was used to install the Firefox and Google Chrome web browser apps. The apps were then launched to ensure that they worked properly. The web browsers were used to navigate and log into a Facebook account and the remote was used to navigate to the page.

4. Forensic Assessment

Various tests were devised and conducted to assess the ability of a digital forensic investigator to acquire an image and identify artifacts of user actions and device information on a Fire TV Stick. Test data was introduced at different times during the testing process. Timestamps were found to be consistent in all the tests; specifically, when files could be retrieved during the assessments, the timestamps were reflective of the simulated interactions with the system. This was confirmed using FTK Imager to triage the extracted information and compare the file timestamps against the recorded times.

4.1 ADB Extraction Test

The `pull` command of the Android debug bridge was used to create an image of the userdata partition without root permissions. The test began by powering on the Fire TV Stick, setting it up, enabling USB debugging and powering off the Fire TV Stick. The device was then disconnected from the TV and connected to a Windows workstation. An Android debug bridge server was then started on the workstation and a connection was established with the Fire TV Stick. The `mount` command was then used to identify the location of the userdata partition. After it was identified, the `pull` command was used to attempt to image the partition, but this failed due to the lack of root permissions. The `dd` command was also attempted, but it failed for the same reason. Thus, the Android debug bridge extraction test failed to extract an image from the device and, therefore, could not help identify any useful information on the device.

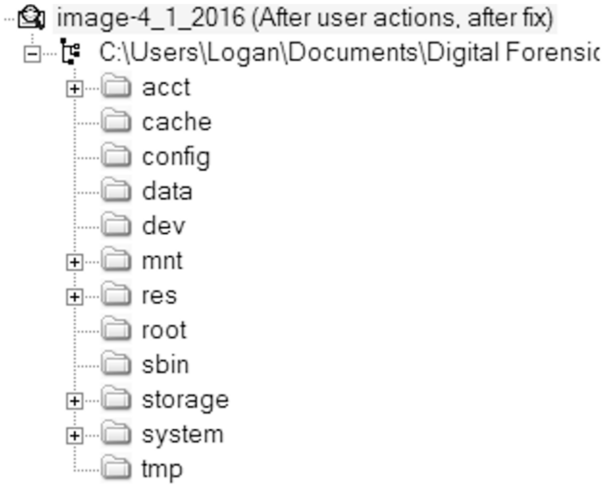
4.2 UFED Touch Test

The UFED Touch v.1.9.0.130 from Cellebrite was used to attempt physical and logical filesystem extractions from the Fire TV Stick. Research by Horowitz [5] has revealed that a physical extraction from the Amazon Kindle Fire HDX could be performed using Cellebrite's UFED Touch Ultimate. Because the two devices use similar operating systems, it was expected that this method could work for the Fire TV Stick.

The experiment involved connecting the Fire TV Stick to the UFED Touch and working through its menus to attempt physical and logical extractions. However, the version of the UFED Touch available at the time of this writing was unable to recognize or read the Fire TV Stick.

4.3 Python Script Test

This experiment attempted to use a custom Python script to extract a logical image of the Fire TV Stick using the Android debug bridge functionality without root permissions. Komodo edit 9, Python 3.5 and the `pyadb` module were used to create the script. The script incorporated native Android debug bridge commands to extract/pull all the files that it could extract, create hashes before and after file transfer (MD5 is



```
image-4_1_2016 (After user actions, after fix)
  C:\Users\Logan\Documents\Digital Forensic
    acct
    cache
    config
    data
    dev
    mnt
    res
    root
    sbin
    storage
    system
    tmp
```

Figure 1. File structure of the test data image produced by the Python script.

available on the Fire TV Stick without any additional modification), compare the hashes, recreate the directory structure and then store all the original timestamps for the files that were extracted. The script was used to create logical images of the Fire TV Stick before and after the test data was added.

FTK Imager v.3.4.0.1 was used to examine the images obtained using the Python script. Figure 1 shows the file structure of the image created after the test data was added. An analysis of the image revealed that some artifacts of user actions were present and could be recovered. The artifacts included remnants of the sideloading process with Kodi, list of installed apps, files and APKs associated with installed apps, app thumbnails and APK files for sideloaded apps. Other system artifacts included the language setting of the device and APK/ODEX files corresponding to background apps.

Certain points regarding the image need to be highlighted. First, many useful data items and artifacts of user actions came from having the ES File Explorer app installed on the device. If this app had not been previously installed on the device, most of what was found in the image would not be present. Second, the data directory, which contains much of the useful user data, could not be extracted due to problems with permissions (i.e., root privileges are required).

4.4 Rooting Test

This experiment was designed to address the problem of not having root permissions to the Fire TV Stick, which hinders access to certain areas of the device. The experiment involved the use of KingRoot v.4.8.5. FireOS on the first-generation Fire TV and Fire TV Stick can be rooted using the KingRoot automatic rooting app [1] up to and including FireOS v.5.0.5. Thus, a copy of the KingRoot APK was downloaded to a workstation and the Android debug bridge was used to install it on the Fire TV Stick. However, the KingRoot GUI is designed to work with a mouse, not the Fire TV Stick remote. Therefore, a Bluetooth mouse had to be connected to the Fire TV Stick to run KingRoot. The Fire TV Stick was successfully rooted and the SuperSU APK was sideloaded using an Android debug bridge install. An Android debug bridge connection was established from the workstation to the Fire TV Stick and the `su` command was executed, successfully gaining root permissions on the device.

After gaining root permissions, the Python script was modified to use root permissions in an attempt to extract files. Initial tests with the modified script revealed that timestamps were not extracted correctly. After changing the operating system on the acquisition workstation from Windows to Linux (Ubuntu), the MAC times were copied correctly.

4.5 ADB Extraction Test

Enabling root permissions on the device reduced the challenges encountered when using the previous Android debug bridge extraction method. Therefore, the test was repeated with root permissions to see if an image could be produced that included the userdata partition.

The procedure involved starting an Android debug bridge server with root permissions to the Ubuntu workstation. A connection was then established to the Fire TV Stick and the Android debug bridge `mount` command was used to locate the userdata partition. The `su` command was executed to gain root permissions. Next, the `chmod` command was used to provide temporary (until reboot) world-read permissions on the userdata block. The Android debug bridge `pull` command was used to successfully extract an image of the Fire TV Stick's userdata partition. Two images were created using this method: (i) test image created initially while working through the empirical process; and (ii) test image after the initial sample data was added.

After adding test data, attempts were made to create another image, but an automatic software update changed the operating system from v.5.0.5 to v.5.0.5.1, rendering the version of KingRoot unable to root

Table 2. Artifacts in the Amazon Fire TV Stick.

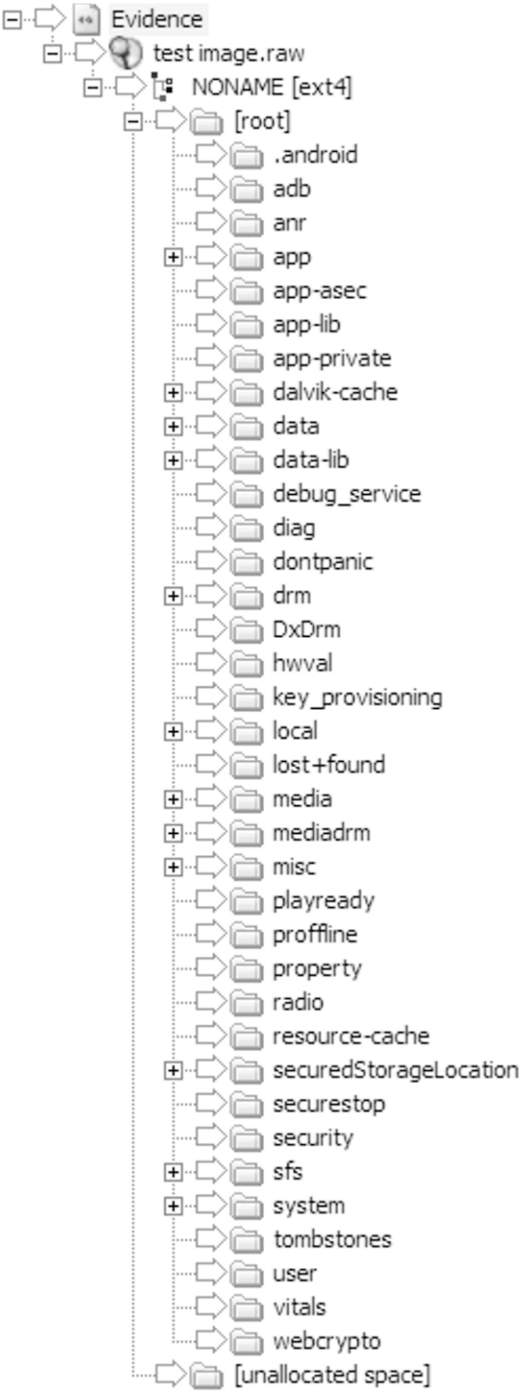
Artifacts	**Description**
Timestamp	Last access time reflective of user interaction.
Browser History	The `browser.db` file contains evidence of navigating to websites using Mozilla Firefox.
Pictures	`[root]/data/com.amazon.bueller.photos/files/cmsimages` contains pictures from the Amazon Cloud Drive. Images extracted directly from the Cloud Drive have the same hash values as the originals, but images found at this location in the Fire TV Stick do not. It appears that images in the Fire TV Stick are formatted for better viewing in the system menu. Two files, each identical in name except for `*-full.jpg` and `*-thumb.jpg` suffixes may be found. Figure 3 shows the original image (left), `fPM452RvROeOv-iKfAaOSQ-full.jpg` (center) and `fPM452RvROeOv-iKfAaOSQ-thumb.jpg` (right).
Bluetooth Devices	`[root]/data/com.amazon.device.controllermanager/databases/devices` contains the names and MAC addresses of devices connected via Bluetooth (Razer Orochi mouse and Amazon Fire TV remote).
Amazon Logs	`[root]/data/com.amazon.device.logmanager/files` contains several log files, including `Log.amazon\main`.

the (at the time of writing) up-to-date Fire TV Stick. The acquisition experiment was halted at this point.

It should be noted that a digital forensic investigator would not put such a device online while working on an active case. The update occurred only because connectivity was required in order to generate the test data needed to assess the Fire TV Stick. However, it was still possible to continue the analysis of the image taken with the initial test data. AccessData Labs v.5.1 was used to analyze the images created using the Android debug bridge `pull` command. Figure 2 shows the file structure of the test image created after the test data was added.

Analysis of the image revealed that a large amount of useful information could be recovered (Table 2). In addition to the artifacts commonly encountered in Android devices, it was also possible to recover several Amazon-specific artifacts.

While the images produced using the Android debug bridge extraction method proved to be extremely useful, a few points regarding the method should be highlighted. First, the method requires permissions to the partition to be changed in order to use the Android debug bridge



```
Evidence
  test image.raw
    NONAME [ext4]
      [root]
        .android
        adb
        anr
        app
        app-asec
        app-lib
        app-private
        dalvik-cache
        data
        data-lib
        debug_service
        diag
        dontpanic
        drm
        DxDrm
        hwval
        key_provisioning
        local
        lost+found
        media
        mediadrm
        misc
        playready
        proffline
        property
        radio
        resource-cache
        securedStorageLocation
        securestop
        security
        sfs
        system
        tombstones
        user
        vitals
        webcrypto
      [unallocated space]
```

Figure 2. File structure of the userdata image obtained via ADB extraction.

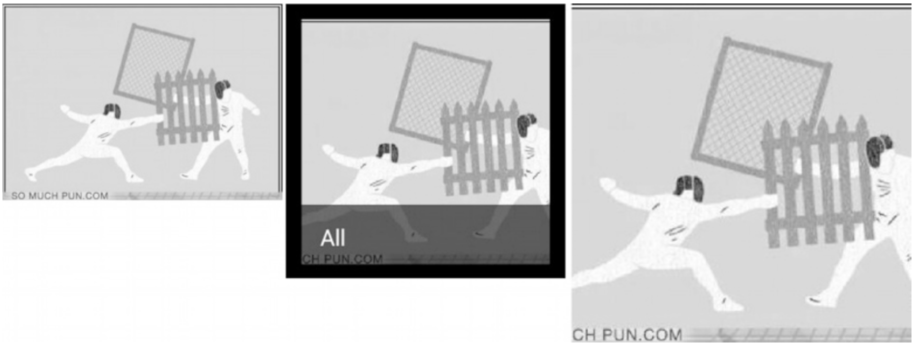


Figure 3. Visual comparison of images obtained from the Fire TV Stick.

`pull` command for extraction. Thus, a change has to be made to the system, which is certainly not ideal with regard to the forensic soundness of the method. Upon closer inspection, the version of KingRoot used was found to have not made significant changes to the userdata partition; however, this would have to be considered for every future root/exploit method. Forensic soundness is still preserved during extraction because the userdata partition is only granted world-read, not world-write, permissions (the permissions are reset after the device is rebooted). Thus, the Android debug bridge cannot modify the data during a `pull` operation.

4.6 Manual Acquisition Test

In order to handle a device that cannot be rooted, additional experiments were performed to elicit artifacts using traditional manual means – specifically, video recordings, photographs and note-taking to capture and analyze the menus visible to a regular user. The test began by powering on the Fire TV Stick and recording the time. The user accessible menus/pages were then examined starting with the Home tab. Each tab/menu was fully documented before proceeding to the next tab. Figure 4 shows a photograph that documents the Home tab of the Fire TV Stick.

An analysis of the videos and photographs revealed that a large amount of useful information could be recovered. Artifacts of user actions that were recovered include:

- Recently accessed apps/content.
- Games downloaded by the user.
- User's gaming profile.

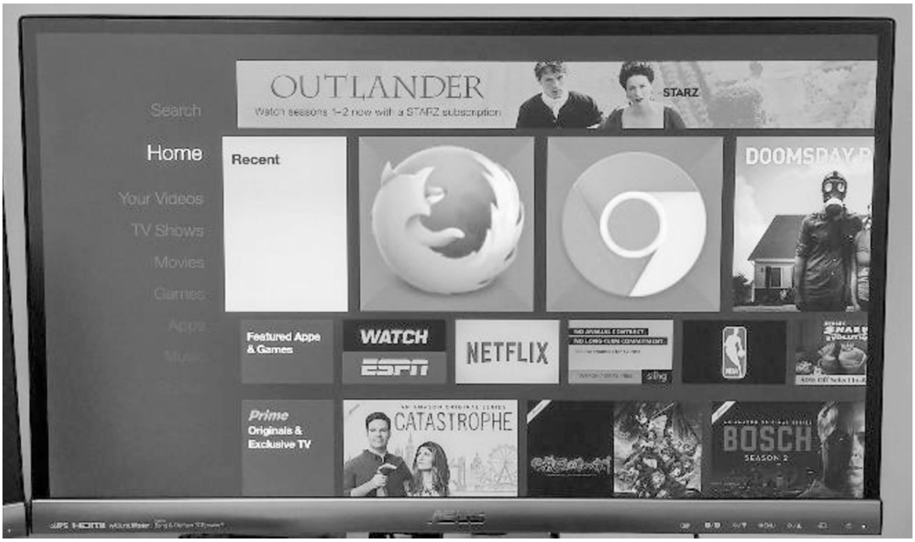


Figure 4. Fire TV Stick Home tab with recent activity feed highlighted.

- List of apps downloaded by the user (sideloaded apps are distinguished by the message "This app was not downloaded from Amazon").
- Amazon Prime music/account content.
- User's Amazon Cloud Drive images/albums.
- Metadata for Cloud Drive images (e.g., name, taken and uploaded timestamps, dimensions).
- Email address associated with the registered user's Amazon account.
- Bluetooth devices synced with the Fire TV Stick.
- Full list of installed apps with metadata (e.g., version, size, storage).
- Name of the registered Amazon account.

Furthermore, the following useful system information was recovered:

- Device name.
- Amazon remote, game controller and other Bluetooth device information (e.g., name, version, serial number).
- Device storage capacity.
- Operating system/software version.
- Device serial number.
- Device date and time.
- SSID of the connected Wi-Fi network.
- Device IP address.
- Wi-Fi adapter MAC address.

- Number of connected controllers/Bluetooth devices.
- System update timestamps.
- Available Wi-Fi networks.
- ZIP code of the location.
- Country and timezone.
- Language settings.

However, this acquisition method is problematic because it requires the device to be analyzed live, which results in changes to the system.

5. Recommended Forensic Analysis Method

Figure 5 outlines the digital forensic methodology recommended for acquiring an image from an Amazon Fire TV Stick.

Step 1 creates the environment for imaging a rooted Fire TV Stick. The Android debug bridge is used to obtain shell access to the device. SuperSU provides root access to the device, enabling all the files to be captured.

Step 2 is the standard best practice for a live investigation. All interactions with the system must be recorded and notes should be taken along with the times in case the device time has been altered.

Step 3 turns the device on. Step 4 sets up the environment variables in the Fire TV Stick to allow the installation of the SuperSU root APK. Step 5 injects the files into the system, establishes root access with the assistance of a Bluetooth mouse (Fire TV Stick does not have a USB port for external peripherals) and confirms that root access is established.

Step 6 uses the Android debug bridge server on the Ubuntu workstation to connect to the Fire TV Stick's shell; the built-in `mount` command is used to identify which partition/block device stores data. Step 7 navigates to the `/dev/block` directory to locate the correct device. Step 8 executes the `su` command on the device to obtain root privileges; following this, the permissions of the userdata partition can be updated to `755`, enabling global read access (but importantly, not global write).

Step 9 exits the Fire TV Stick shell and, given the temporary changes made to the userdata area, maximizes data acquisition via the Android debug bridge `pull` command. Step 10 concludes the data acquisition, powers off the Fire TV Stick and video capture device, and records the times of both actions.

6. Conclusions

An Amazon Fire TV Stick contains a plethora of information, some of it related to user activity and other information related to the system

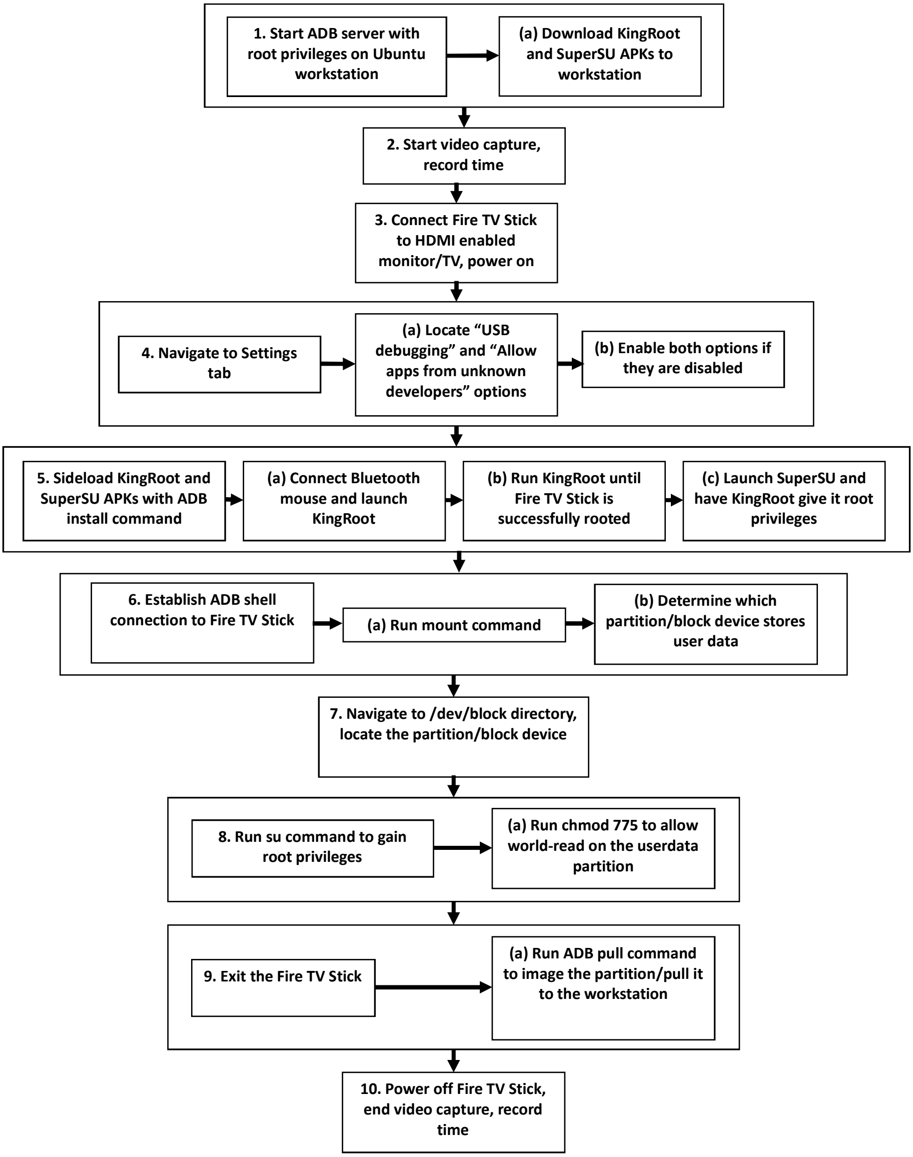

1. Start ADB server with root privileges on Ubuntu workstation
(a) Download KingRoot and SuperSU APKs to workstation
2. Start video capture, record time
3. Connect Fire TV Stick to HDMI enabled monitor/TV, power on
4. Navigate to Settings tab
(a) Locate "USB debugging" and "Allow apps from unknown developers" options
(b) Enable both options if they are disabled
5. Sideload KingRoot and SuperSU APKs with ADB install command
(a) Connect Bluetooth mouse and launch KingRoot
(b) Run KingRoot until Fire TV Stick is successfully rooted
(c) Launch SuperSU and have KingRoot give it root privileges
6. Establish ADB shell connection to Fire TV Stick
(a) Run mount command
(b) Determine which partition/block device stores user data
7. Navigate to /dev/block directory, locate the partition/block device
8. Run su command to gain root privileges
(a) Run chmod 775 to allow world-read on the userdata partition
9. Exit the Fire TV Stick
(a) Run ADB pull command to image the partition/pull it to the workstation
10. Power off Fire TV Stick, end video capture, record time

Figure 5. Forensic analysis method for the Amazon Fire TV Stick.

itself. The proposed method for imaging Fire TV Sticks enables digital forensic investigators to perform analyses of these popular streaming media devices. Efforts are taken to minimize, if not eliminate, data alteration. Thus, the method can be considered to be "semi" forensically sound.

Whether or not a particular Fire TV Stick can be imaged successfully using the proposed method depends on the operating system/software version. It is possible to use KingRoot to root a Fire TV Stick device that runs a Fire OS version earlier than v.5.0.5.1; rooting the device makes it possible to acquire an image using the proposed method. A device running Fire OS version v.5.0.5.1 or later cannot be rooted using the current version of KingRoot and, thus, an image of the device cannot be extracted via the proposed method. An automatic Fire OS update increases the potential of eliminating root access to a device, making it imperative to ensure that the update server is blocked by a firewall or the forensic analysis of the device is conducted in a Faraday cage.

Downgrading the Fire TV Stick software/firmware may make the device rootable using KingRoot. Future research will investigate this possibility as well as the potential effects on the data stored in the device.

The Fire TV Stick has a remote app, a companion application provided by Amazon, which enables the device to be controlled by a smartphone. It provides voice search, navigation, playback control and keyboard text entry features. Future research will analyze the interactions between the remote app and Fire TV Stick to determine if any forensic artifacts are retrievable.

Meanwhile, new streaming services and applications are emerging as streaming media devices become increasingly popular. Future research will also examine these services and applications, which may provide artifacts of interest to digital forensic investigators as well as new avenues for analyzing Fire TV Sticks.

References

[1] AFTVnews, Fire OS 5 on the Amazon Fire TV 1 and Fire TV Stick can be rooted, February 20, 2016.

[2] T. Cushing, Amazon Fire TV firmware update bricks rooted devices, prevents rollback to previous firmware versions, *Techdirt*, December 5, 2014.

[3] B. Evangelista, Cord cutting accelerated in 2015, on track to continue next year, *San Francisco Chronicle*, December 31, 2015.

[4] K. Fairbanks, An analysis of Ext4 for digital forensics, *Digital Investigation*, vol. 9(S), pp. S118–S130, 2012.

[5] J. Horowitz, Kindle Fire HDX Forensics (`kindlefirehdxforensics.blogspot.com`), April 15, 2014.

[6] A. Iqbal, H. Al Obaidli, A. Marrington and I. Baggili, Amazon Kindle Fire HD forensics, *Proceedings of the International Conference on Digital Forensics and Cyber Crime*, pp. 39–50, 2014.

[7] J. Smith, Here's why consumers are increasingly turning to streaming media devices to view content, *Business Insider*, June 16, 2016.

[8] N. Terry, Amazon Fire TV takes 30% of the streaming market, *Android Headlines*, June 5, 2015.

[9] P. van Bolhuis and C. Van Bockhaven, Forensic Analysis of Chromecast and Miracast Devices, Cybercrime and Forensics Project, Master's Program in System and Network Engineering, University of Amsterdam, Amsterdam, The Netherlands, 2014.