AUTOMATED AUTHENTICATION OF EXEMPLAR MEDIA IN A DATABASE

By

BRENT M LARSEN

B.S., University of Colorado, 2013

A thesis submitted to the

Faculty of the Graduate School of the

University of Colorado in partial fulfillment

Of the requirements for the degree of

Masters of Science

Recording Arts

2016

This thesis for the Master of Science degree by

Brent Michael Larsen

has been approved for the

Recording Arts Program

by

Catalin Grigoras, Chair

Jeffery Smith

Jason Lewis

April 29, 2016

Larsen, M. Brent (M.S. Recording Arts)

Framework for Automating Authentication of Digital Media Files for an Exemplar Database

Thesis directed by Professor Catalin Grigoras

ABSTRACT

The focus of this work is to document the creation of a database containing original audio recordings and images. Content for the database is managed through a web based application in which information can be passed to the database from the user. A fundamental feature of the media database application is the automated verification or rejection of a file based on its originality. By performing a string search of the file's metadata, signs of non-originality due to prior processing from a software editor can be detected thus preventing the file from being added to the database. Additionally, the media database includes user restricted access for both submissions to and retrieval of database information.

The form and content of this abstract are approved. I recommend its publication

Approved: Catalin Grigoras

TABLE OF CONTENTS

LIST OF FIGURES

FIGURE

CHAPTER I

INTRODUCTION

Purpose

Due to the ever changing nature of digital media, the field Digital Forensics must be responsive in staying current with new digital devices. One method that can be employed in the aiding of digital examiners is the use of reference media otherwise known as exemplars. Exemplar media can provide valuable metadata information about the device on which a media file was created. By building a large collection of exemplars an examiner can use these references against a media file of unknown origin to help identify any key similarities. In order for this method to be effective a large collection of authentic original files must be maintained. A database can be valuable resource once the number of exemplars grow and a need for easy access is present. A properly implemented database can also prove useful in adding of media created on newer digital devices, thus addressing this need to staying up-to-date with digital devices.

In order to meet the needs of the intended user a media database should be efficient in its operation. In our proposed exemplar database, the challenge of adding new media to the in which new exemplar media files are added must be simplified. Authenticating a large quantity of media files can be an arduous task for a single individual to take on. In this instance much of the work can be divided up by automating a base-level of authentication and incorporating a multiple user submission method. In addition to reducing the amount of time it takes to submit a media file, automating authentication also serves the purpose of acting as a gatekeeper for the media entering the database. If a submitted file is determined to be consistent with that of an original, it is permitted to enter into the database. If the file appears to be inconsistent with that of an original, it is rejected.

Authentication

One of the critical components this database relies upon is that the media contained within must be authentic to have any inherent value. In determining whether a file is authentic what we are looking for is that a file is consistent with the operation of the recording device used to make the media. [1] One method of analysis for authenticating a digital media file is by evaluating its file structure. A digital media file is comprised of two parts. One part is the actual media that represents the audio, video, or image and the other part contains information about the audio, video or image. In Scott Anderson's Analytical Framework for Authenticating Digital Images he likens the digital file to a can of soup. In his analogy the soup within the represents the actual media, the can the container in which the media is stored, and the label being information about the media. [2]

This metadata that is stored inside a file gives instruction about how the file is assembled, the contents of the file, information about the media, and information about the file. A key feature within this metadata is the Exchangeable Image File Format or EXIF. The EXIF of a media file may include specific information on the creation date and time, the device information on which the file was created, f-stop and ISO speeds for images, sample rate and number of channels for audio, along with other relevant information about the media.

A common method in assessing a file's structure is by viewing it as hexadecimal notation data. A hex viewer is a valuable tool that converts the raw binary data into a more human-readable format. The converted text itself is for the most part unintelligible but certain bits of EXIF information may prove easier to interpret [Figure 1]. In addition to a media device populating the metadata for a digital file, alterations to the file may be present as a result of a software editor. For example, we could evaluate the metadata of an mp3 audio recording created by a Tascam GT-R1 portable recorder and could expect find information about the device on which it was recorded [Figure 2]. However, if this mp3 file was then opened in a software editor such as Adobe Audition and then saved with the exact name we would see that new information was added to the metadata [Figure 3]. Whether any editing of the mp3 takes place not, Adobe Audition adds its own creation information into the file. At this point it is upon this criterion that we must no longer consider the mp3 an original recording as it no longer consistent with the operation of the device on which it was created.

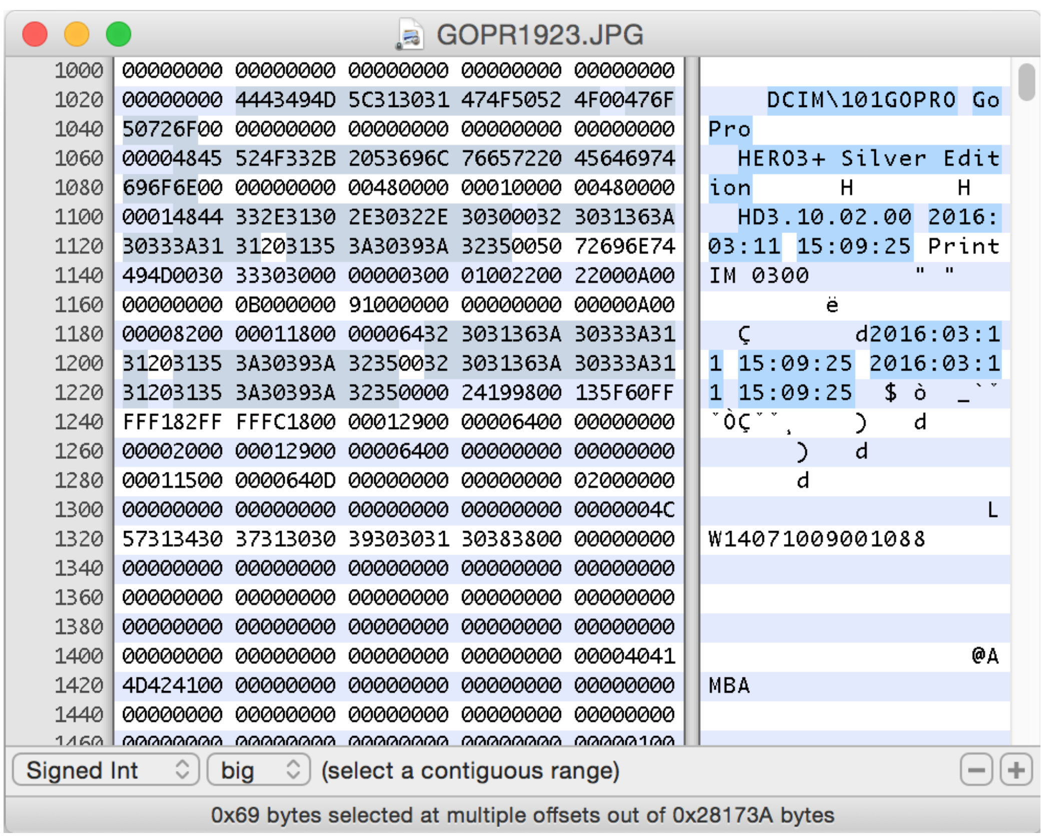

FIGURE 1: Example of EXIF information displayed in a hex viewer

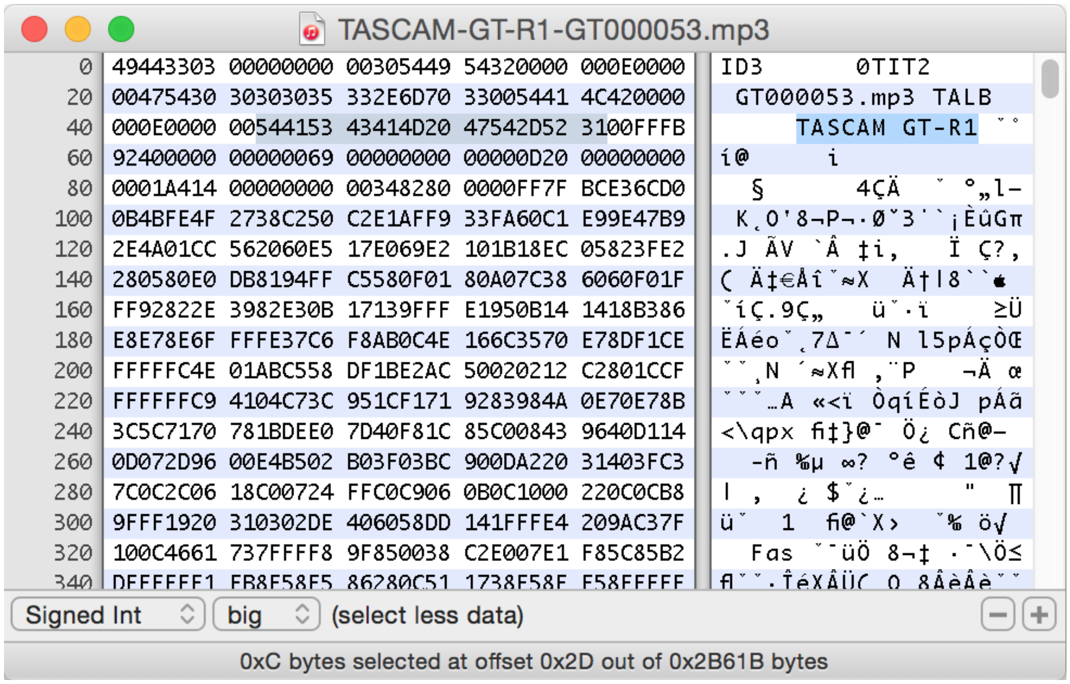

FIGURE 2: EXIF device information of an original audio file in hex viewer

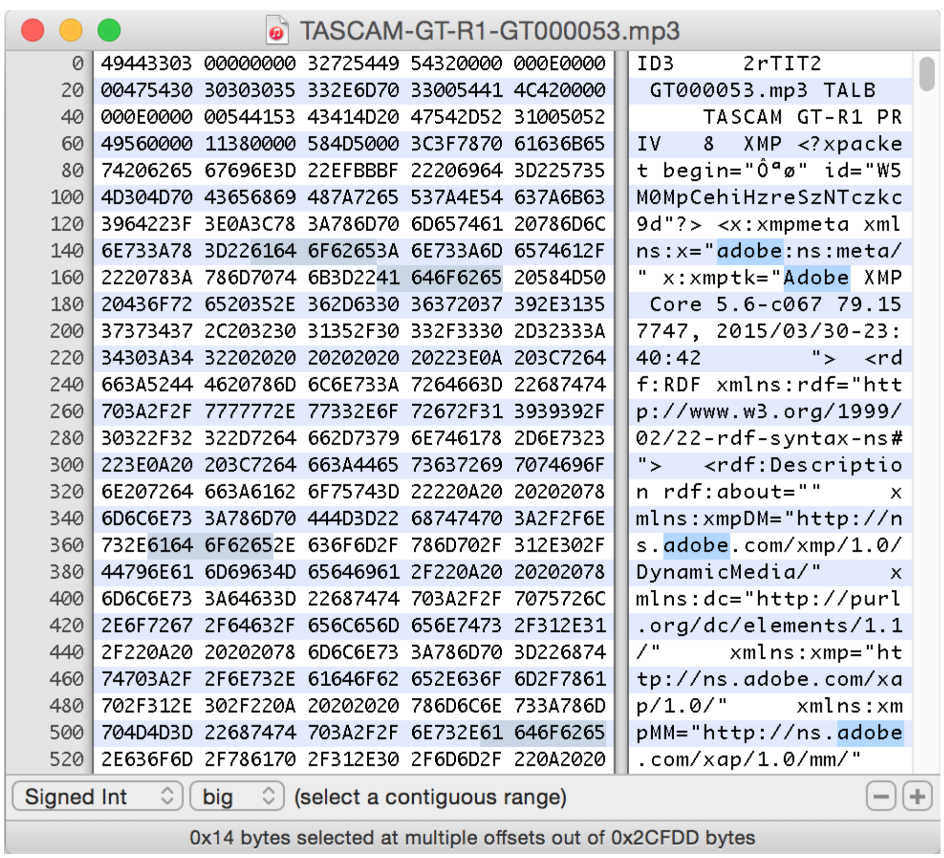

FIGURE 3: EXIF of audio file from FIGURE 2 after being saved in Adobe Audition

Methods for Authentication using Python

By scripting the analysis of a media file, the authentication process is sped up and any potential human error can be mitigated. Python modules allow for a customizable set of tools that can perform specific tasks with a very high rate of efficiency. For example, with a few lines of code we can analyze the EXIF of a jpg image with the help of the exifread module [Figure 4]. This code performs the basic function of opening a file, utilizing the the exifread module to extract EXIF tag information, and then outputting the findings into a text file [Figure 5].

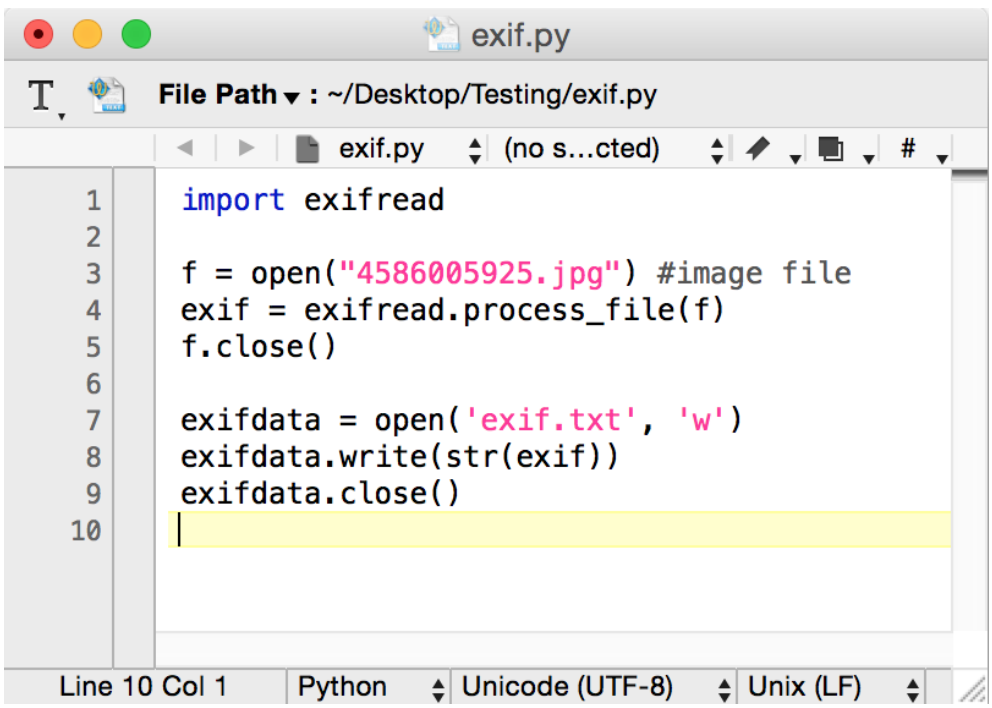

FIGURE 4: Python exifread operation

exif.txt

```
ExposureTime    (1, 320)
FNumber (8, 1)
CustomRendered  0
MaxApertureValue        (3, 1)
FocalLength     (61, 1)
ShutterSpeedValue       (8321928, 1000000)
Flash   16
WhiteBalance    0
ExifOffset      70
MeteringMode    5
DateTimeDigitized       2015:07:09 08:58:59
DateTimeOriginal        2015:07:09 08:58:59
ApertureValue   (6, 1)
ISOSpeedRatings 100
Model   Canon EOS 5DS
Make    Canon
ExposureMode    1
ExposureBiasValue       (0, 1)
```

FIGURE 5: Text output of EXIF organized

At this point, the information generated with Python still produces results that must be manually examined. We can further automate this process by parsing the file and instructing Python to interpret the results for us. The next process we can apply is a string search of the media file's metadata in order to look for traces left by software editors. Because relevant EXIF data occurs at the beginning of a digital file, we can expedite the search process by limiting the search to the first and last 5,000 bytes of the file. Figure 6 shows the output of an original image made with a Nikon D3300 camera that has been opened by Python, and then the first 5000 bytes are converted to ASCII and outputted into the terminal. [Figure 7]

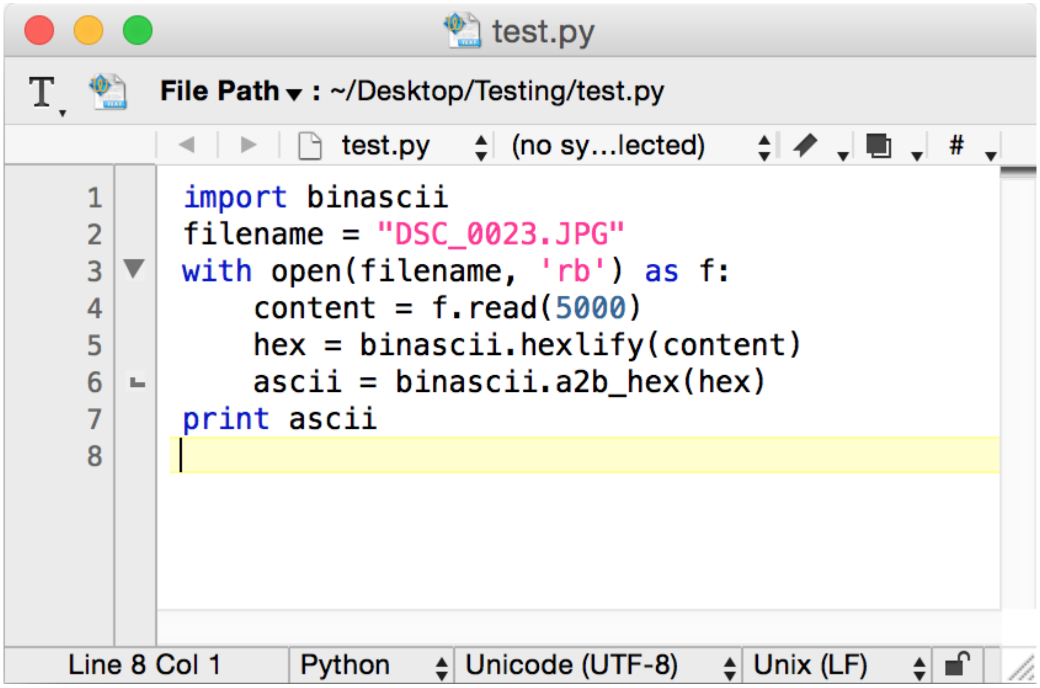

FIGURE 6: Python script for converting first 5000 bytes to ASCII

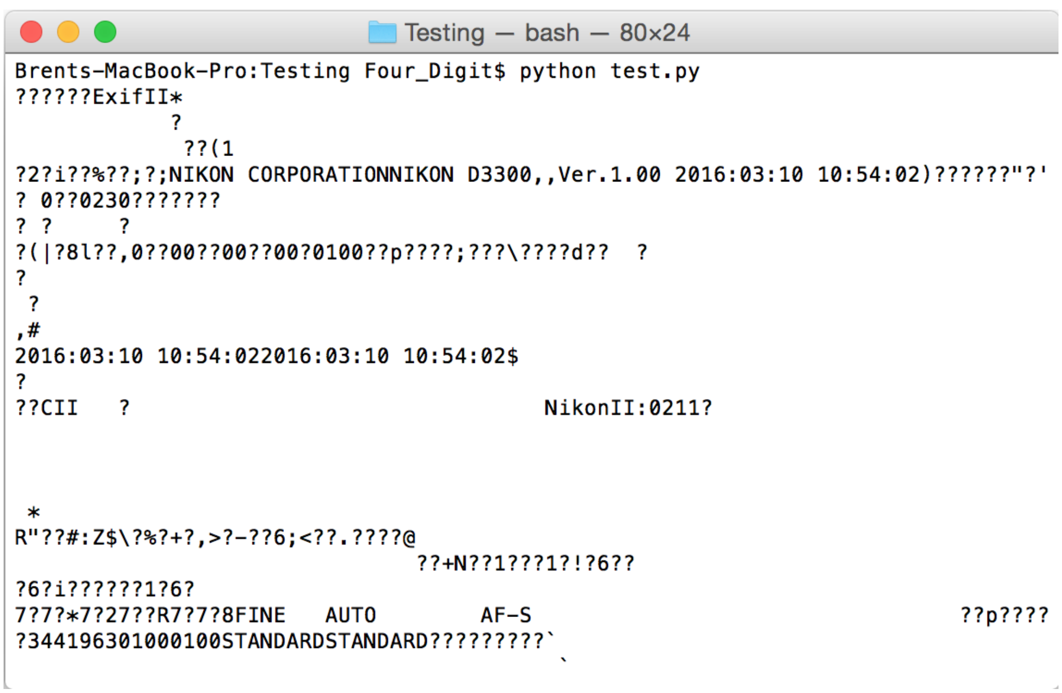

FIGURE 7: Terminal output of the first 5000 bytes in ASCII of image

These results can then be organized into a more human-readable format, however for automation purposes this is not pertinent. For the next step of this process a dictionary of commonly found words left in metadata by software editors is created. Figure 8 shows an example dictionary of “black listed” words that are commonly left by image viewing and editing software. Python then takes this dictionary and searches its contents against the contents of the ASCII string that was converted from binary in the previous steps. If any of the black listed words are found within the media file, Python is then instructed to make a decision based on its findings.

words_black_list.txt

```
Adobe
adobe
AppleMark
Corel
corel
GIMP
Gimp
gimp
http://www.iec.ch
JFIF
Lightroom
picasa
Picasa
Photoshop
quicktime
Quicktime
QuickTime
Windows
windows
```

FIGURE 8: Sample of blacklisted word dictionary

CHAPTER II

GUIDELINES FOR MULTIPLE USER CONTRIBUTIONS

Crowdsourcing materials for use in any scientific research comes with the risk of introducing a margin of error to the results. In allowing multiple users to contribute new media to the database it can be said that a level of uncertainty may be introduced regarding a file's originality. By implementing specific control measures this level of uncertainty can be reduced. Four proposed methods of ensuring the integrity of the database remain intact include:

1. Users accounts are assigned to the user by administrators of the database. Those who wish to gain access to the database must obtain their credentials by submitting a request to administrators.
2. Authentication of media being submitted to the database is performed automatically. This is to ensure that potential error or bias is mitigated.
3. When a user successfully submits a file to the database their user id and time of submission is also added to the database entry. This will provide a record in the case that any suspicious activity occurs.
4. If a user attempts to upload a media file that is determined not to be authentic they will be given warning. A limited number of infractions will be tolerated before the privileges of the users will be suspended.

CHAPTER III

DATABASE FRAMEWORK

Front and Backend Structures

Having defined the basis on which media files are to be authenticated, this section provides documentation of the development of a working media exemplar database for use by the National Center for Media Forensics. The database is built around a web accessible application in which authorized users can access and/or upload new media depending on their privileges.

The front end of the application is handled by Flask. Flask is Python micro-framework which provides tools for creating a web based application around the Python language. [3] Flask provides several benefits including database integration as well as the ability to incorporate advanced Python modules into Flask. The backend of the database is handled by MySQL. MySQL is an open-source relational database management system that handles client-server responsibilities. It is also one of the most widely used database management system and is effective in communicating with Flask. [4]

User Identity Management

Even though MySQL and Flask are fully capable of handling User Identity and Access Management, an additional layer of security can be added by utilizing an outside user authorization service. User authorization service for this database are handled by Stormpath. The decision to choose this service was based upon the ease of implementation within

Flask's framework as well as high standard of security and best practices that they adhere to. [5] Many of these best practices followed by Stormpath can be observed outside of their services, however it should be noted that development deadlines helped form this decision.

Stormpath allows for administrative control over user accounts to be managed through a web-based format [Figure 9]. User access controls can be managed at any time to define whether a specific user is limited to access only privileges or if they have permission to contribute original files to the database. [Figure 10] Stormpath also handles a number of other account management features key to ensuring a database's success such as user password rest and email communications.

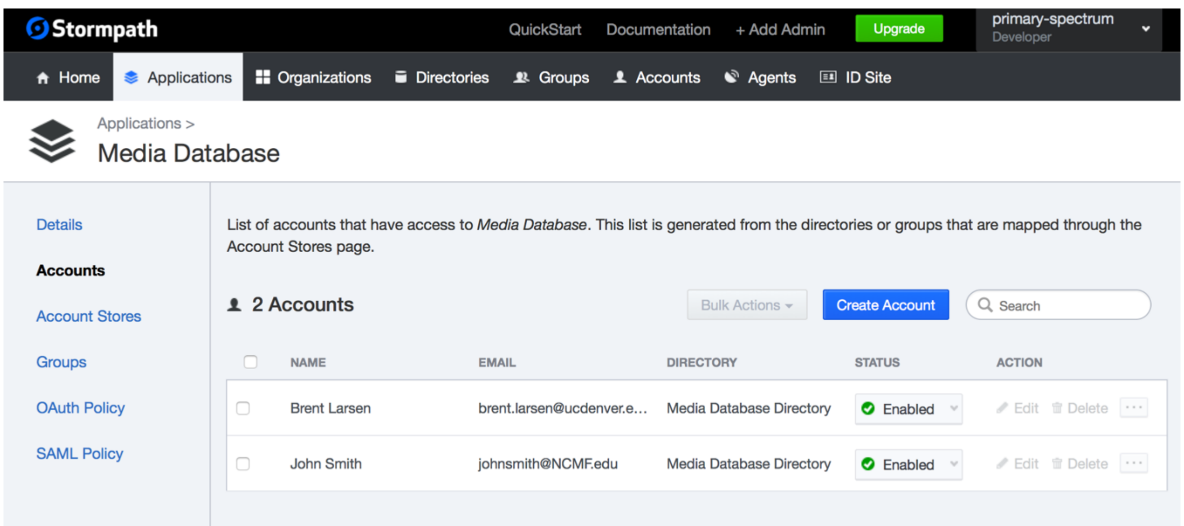

FIGURE 9: Stormpath user account management page

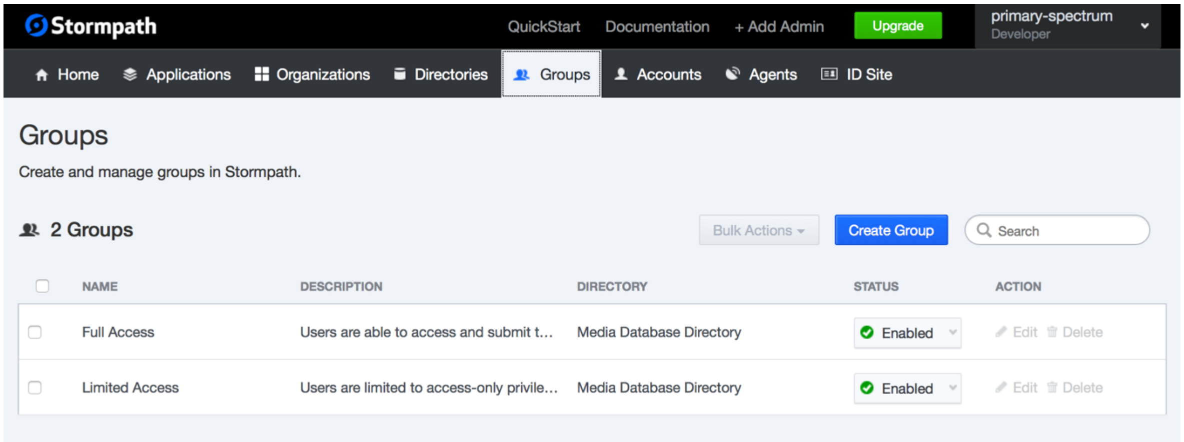

FIGURE 10: Stormpath user group management page

CHAPTER IV

TOOL VALIDATION METHODS & MATERIALS

The Scientific Working Group on Digital Evidence calls for the validation of, "all tools, techniques and procedures utilized in the performance of digital forensics." [6] Validation of the database is key in ensuring proper functionality and that repeated use will yield similar results. The primary function that this testing addresses is the ability of the automated authentication to detect media that is not considered original.

Because the database accepts audio, video, and image files it was important to test each file type to ensure proper validation. Five different devices from each category were selected to provide the original media content.

1. Handheld audio recorders used:

   Olympus DM-620

   Tascam DR-07

   Zoom H1

   Sony ICD-SX750

   Tascam DR 100mkII

2. Digital Cameras used (images):

   Panasonic DMC-FS7

   Casio EX-Z150 [7]

   Sony DSC H50 [7]

   Nikon D200 [7]

   Canon EOS 30D

3. Cameras used (video):

Nikon CoolPix L18

Sony Cybershot DSC-S650

GoPro Hero 3+

Canon PowerShot Pro1

Olympus EPL1

The media created on these digital devices were then manually authenticated by performing a metadata analysis Copies of these files were then made and each was opened within various software editors. Each file was then resaved and given the same name. Again metadata analysis was performed on the newly created files to manually identify the traces left by software editors. In order to test the function of the database an admin user was logged into the database application and each file was attempted for submission.

CHAPTER V

RESULTS

Of the 15 original media files submitted to the database, each file passed the requirement determined by the algorithm to successfully enter into the database. Figure 11 is a view of the MySQL database table containing the of the 15 media files.

Result Grid Filter Rows: Search Edit: Export/Import:

media_id	media_user_id	media_type	media_make	media_model
1	brent.larsen@ucdenver.edu	audio	TASCAM	DR-100MK2
2	brent.larsen@ucdenver.edu	audio	TASCAM	DR-07
3	brent.larsen@ucdenver.edu	audio	ZOOM	H1
4	brent.larsen@ucdenver.edu	audio	SONY	
5	brent.larsen@ucdenver.edu	audio	OLYMPUS	DM620
6	brent.larsen@ucdenver.edu	video	OLYMPUS	E-PL1
7	brent.larsen@ucdenver.edu	video	CANON	
8	brent.larsen@ucdenver.edu	video	SONY	DSC-S65
9	brent.larsen@ucdenver.edu	video	NIKON	L18
10	brent.larsen@ucdenver.edu	video	GoPro	
11	brent.larsen@ucdenver.edu	image	Panasonic	DMC-FS7
12	brent.larsen@ucdenver.edu	image	CANON	EOS 30D
13	brent.larsen@ucdenver.edu	image	CASIO	EX-Z150
14	brent.larsen@ucdenver.edu	image	NIKON CORPORATION	D200
15	brent.larsen@ucdenver.edu	image	SONY	DSC-H50

tbl_media 1

FIGURE 11: MySQL table containing successful media file submission

Each of the 15 media files that were opened and then resaved in various software editors and queued for submission were rejected based upon traces of software editors present. The algorithm determined in each case that the newly saved media files contained traces left by software editors.

CHAPTER VI

CONCLUSION

Discussion

The documentation of the development of an exemplar media database framework detailed in this paper outlines a practical method for the authentication of audio, video, and images created on digital devices. In addition, validation testing of the proper functioning of this database ensures that it can be considered a tool that is both reliable and reproducible in its results. It should be noted that keeping the blacklist word dictionary is an important task that will need to be regularly evaluated.

Future Research

This exemplar database was designed to so that it could be expanded upon in the future to incorporate additional features. The current abundance of Python modules that are openly available means that further customization and additions can be made in order to meet the needs of the database. It could also be said that the overall scope of the database could be redefined as new features are added. Currently the database serves the purpose of providing reference material aiding with the comparisons against unknown media, but potential future features worth consideration include:

- Structure Analysis
- Quantization Table extraction from cameras [8]
- Clone Detection for images [9]
- Photo Response Non Uniformity (PRNU) mapping [10]
- Transition and zero level analysis of audio recordings

REFERENCES

[1] Scientific Working Groups on Digital Evidence and Imaging Technology (May 27, 2015). SWGDE and SWGIT digital & multimedia evidence glossary v2.8. Retrieved March 14, 2016, from the Scientific Working Group on Digital Evidence website: https://www.swgde.org/documents/Current%20Documents/

[2] Scott Dale Anderson (2011). Digital Image Analysis: Analytical Framework For Authenticating Digital Images. University of Colorado Denver, 18

[3] Ronacher Armin (2013). Flask's Documentation. Retrieved March 2, 2016, from Flask website: http://flask.pocoo.org/docs/0.10/

[4] Oracle (2016) Oracle Products and Services Overview Retrieved March 17, 2016, from Oracle website: http://www.oracle.com/us/products/mysql/overview/index.html

[5] Stormpath (2016) Flask-Stormpath Documentation. Retrieved March 14, 2016, from Stormpath website: https://flask-stormpath.readthedocs.org/en/latest/index.html

[6] Scientific Working Group on Digital Evidence (September 5, 2014). SWGDE Recommended Guidelines for Validation Testing v2.0. Retrieved March 15, 2016, from the Scientific Working Group on Digital Evidence website: https://www.swgde.org/documents/Current%20Documents/

[7] Gloe, T., & Böhme, R. (2010). The Dresden Image Database for benchmarking digital image forensics. In Proceedings of the 25th Symposium on Applied Computing (ACM SAC 2010) (Vol. 2, pp. 1585-1591)

[8] Christy, John (1992). Read/Write JPEG Image Format. Retrieved April 26, 2016 from Github: https://github.com/ImageMagick/ImageMagick/blob/05d2ff7ebf21f659f5b11e45afb294e152f4330c/coders/jpeg.c - L818

[9] Vasiliauskas Agnius (2009). Detecting Copy-move Forgery in Images. Retrieved April 27, 2016 from Vasiliauskas' blog: http://coding-experiments.blogspot.com/2009/03/detecting-copy-move-forgery-in-images.html

[10] Jocelin Roasles Corripio, Ana Lucila Sandoval Orozco, Luis Javier Garcia Villalba, Julio Hernanzez-Castro, Stuart James Gibson (2013). Source Smartphone Identification Using Sensor Pattern Noise and Wavelet Transform. University of Kent. Retrieved April 27, 2016 from: https://kar.kent.ac.uk/37195/1/corripio2013.pdf