BitCuratorEdu Learning Object

BitCurator Software: Creating a Disk Image Using Guymager

Discussion questions to pair with the screencast

Authors 1

Description 1

Learning object type 1

Learning objectives 1

Screencast 2

Discussion Questions 2

Answer Key 5

Tools and Resources Mentioned in This Document 13

Authors

Cal Lee, Hannah Wang

Description

These discussion questions can be used to encourage student engagement with the BitCurator screencast, Creating a Disk Image Using Guymager. The questions can also be used for discussion accompanying a live demonstration, a guided hands-on exercise, or independent exploration of the BitCurator Environment.

Learning object type

Lesson plan/materials

Learning objectives

BitCuratorEdu Learning Object

This learning object might be used in a lesson to satisfy the following learning objectives:

- Identify the appropriate tools to: safely acquire born-digital materials from storage media and other modes of transfer; assist in the appraisal of born-digital materials; scan for sensitive information in born-digital materials; and package born-digital materials for preservation and access.
- Articulate the relationship between actions taken during acquisition and processing of born-digital materials and their long-term preservation and access.

Screencast

https://youtu.be/gvkuQwdMRYc

Discussion Questions

These discussion questions can be used to encourage student engagement with the BitCurator screencast linked above. The questions can also be used for discussion accompanying a live demonstration, a guided hands-on exercise, or independent exploration of the BitCurator Environment. Video timestamps are included in parentheses, where applicable.

1. At what point(s) in a digital curation workflow might you expect to use Guymager?

2. When mounting an external data storage device to your computer, what are the differences between (1) creating a “disk image” of the data on the device and (2) copying the files from the device to your computer?

3. What types of information are present in a disk image but not in the files themselves?

4. Why might you want to create a disk image rather than only copying the files?

5. Why might you not want to retain the image of a device over the long term?

6. What does it mean to “set the mount policy to read-only” (0:59)? Why would you do this?

7. What are the differences between setting the mount policy to read-only in the BitCurator environment (0:59) and using a hardware write blocker?

8. Why does Guymager show so many different devices (2:54)? Are these all physical devices attached to the host computer?

9. What are the differences between a raw (dd) disk image and an Expert Witness Format (.E01) disk image (3:14)?

10. What factors might you consider when deciding whether to create a raw or EWF disk image (i.e., what are the advantages and disadvantages of each)?

11. How might you use the optional metadata fields in Guymager (3:30)? Would you prefer different labels for those fields? If so, what might they say instead?

12. Notice that Guymager has a setting for “split size.” (3:15-4:50). What does that do, and why would you want to split a disk image?

13. Guymager has a setting for checksum/hash calculation and validation (4:12-4:46). [Note: It can be helpful to point out that, while the two terms have somewhat different meanings, most people use the terms “cryptographic hash” and “checksum” interchangeably.] What is a checksum/hash?

14. Why might you want to generate and validate checksums?

15. Does generating a checksum for a file ensure that the file is an authentic record? Why or why not?

16. What’s the difference between selecting “Calculate MD5,” “Calculate SHA-1,” and “Calculate SHA-256”?

17. How would you decide between using MD5, SHA-1, or SHA-256 or a combination of them?

18. Guymager provides various types of information in the .info file (in this case image1.info). (5:54-7:34) How would you characterize the types of information you can find in the .info file?

19. Which information from the .info file would want to retain over time? Why? Who would you see using the information, and how would you see them using it?

20. The .info file has a MD5 hash and verified MD5 hash for the disk image (6:47-7:07). A hash is a specific output based on a specific input. What was the input to calculate this MD5 hash?

21. If I were to run a checksum tool against the file called image1.E01 (5:31), would I get the same result MD5 hash as the one stored in the .info file? Why or why not?

Answer Key

These discussion questions can be used to encourage student engagement with the BitCurator screencast linked above. The questions can also be used for discussion accompanying a live demonstration, a guided hands-on exercise, or independent exploration of the BitCurator Environment. The questions **(in bold text)** *ask students to analyze the social and technological context of the BitCurator Environment and the tools packaged in the distribution. Example answers are given* (in regular text)*, though some questions are subjective and answers may vary, depending on the knowledge of the student and the scope of the class. Video timestamps are included in parentheses, where applicable.*

At what point(s) in a digital curation workflow might you expect to use Guymager?
Students could provide a variety of answers. They should consider the places in a workflow when someone might want to create a disk image or view the "device info" for a storage medium.

When mounting an external data storage device to your computer, what are the differences between (1) creating a "disk image" of the data on the device and (2) copying the files from the device to your computer?
A disk image is an exact copy of all the storage sectors on a device, so it replaces all of the information from the filesystem (e.g., file permissions, timestamps, directory paths) as well as hidden files and anything in unallocated space (from deleted files). It's essentially the disk (all of the bits) without the disk (no longer relying on the original hardware). By contrast, a standard file copy just moves the individual files from one place to another.

What types of information are present in a disk image but not in the files themselves?

- Filesystem metadata
- Contents of unallocated space (often including data from delete files)
- Hidden files and other system files
- Low-level structures such as the master boot record

Why might you want to create a disk image rather than only copying the files?

- Make sure the full set of bits is safe (allows you still to have the disk but not have to depend on a fragile physical medium).
- There may be surprises within the structure of the file system (e.g. hidden files)
- You could inadvertently change something in the act of examining or dealing with the files, e.g., byte order, character encoding, filesystem metadata, or hidden files
- Proof of file integrity and chain of custody - If there are questions about whether a given source was the basis for a given set of digital objects, one can go back to the original bits and compare hash values.
- Corrupted files and viruses - Having the whole bitstream available (in a controlled and safe staging area) makes it possible to determine what subset of the bitstream can actually be recovered in a useful way.
- There are likely to be changes in preservation strategy or access conditions over time. Default ingest process is to create a normalized AIP from a given type of SIP (e.g. convert all Word documents to PDF). This is almost certain to lose some information in the process. Future techniques or access scenarios might require access to the original Word files. There can also be important information embedded in the filesystem.
- Embedded Contextual Information and User Artifacts - Depending on understanding of arrangement with the Producer, hidden data can also serve as important evidence for the curation of a collection, e.g., traces of data that indicate what application created the files, login, or password information that's necessary for accessing various data sources. For further discussion of possibilities, see: Garfinkel, Simson, and David Cox.
- If you'd like to access materials through emulation, disk images can be an essential ingredient.

Why might you not want to retain the image of a device over the long term?

- Because a disk image includes the contents of all sectors on a disk, it can require more storage space. However, disk images often compress very well, because they tend to include a lot of empty space (a series of zeros is easy to compress), which can help to address this storage issue.

- If the disk was just a carrier and all that matters is the payload (content) of files, there's no need to create a disk image.
- If there is problematic data on the disk (e.g. sensitive data, malware), one may decide to retain only the individual files, rather than keeping the full disk image.

What does it mean to "set the mount policy to read-only" (0:59)? Why would you do this?

This means that the user can see and access the files and folders on the disk, but it's not possible to write to (i.e., change) the disk. Note that it's even better to use a hardware write blocker, but changing the mount policy can help to avoid some accidental changes. It can be a good idea to set the mount policy as read-only when attaching storage devices that include primary sources to image or examine their contents.

What are the differences between setting the mount policy to read-only in the BitCurator environment (0:59) and using a hardware write blocker?

Changing the mount policy is telling the operating system not to allow any writes to the attached storage device. By contrast, a hardware write blocker is a physical device that interfaces between the operating system and the storage device. Hardware write blockers work independently of the operating system and usually have better visual indicators to confirm that your computer is not writing to the storage device.

Why does Guymager show so many different devices (2:54)? Are these all physical devices attached to the host computer?

Only the first item on the list (Memorex USB_Flash_Drive) is a physical device connected to the host computer. The second item (VBOX_HARDDISK) is a virtual disk (see: https://www.virtualbox.org/manual/ch05.html#vdidetails); this is VirtualBox tricking the Linux guest operating system into thinking that it has its own physical storage device. All of the other items on the list are loop devices, which are actually individual files that Linux makes available as if they were storage devices (see https://en.wikipedia.org/wiki/Loop_device). You'll see that all of the loop devices have names in the Model column that end with .snap. SNAP is a way of installing software in Linux (see https://en.wikipedia.org/wiki/Loop_device); every SNAP software package appears as a loop device that has a virtual filesystem (SquashFS).

What are the differences between a raw (dd) disk image and an Expert Witness Format (.E01) disk image (3:14)?

A raw disk image is simply a copy of all of the contents of the storage sectors, written out to a single file or set of files (can be split into smaller chunks to make them more manageable and so that the resulting images can fit onto limited filesystems and media such as FAT or DVD/CDROM). There is no additional metadata embedded in the file.

What factors might you consider when deciding whether to create a raw or EWF disk image (i.e., what are the advantages and disadvantages of each)?

- Raw (dd): Advantages include that it's very simple, and you can use relatively simple tools to mount and manipulate the image; the Image can be easily split for storage and transport on removable media; and output can be piped to other applications for immediate processing. Disadvantages include that the image can be very large because data aren't compressed (can be addressed by applying compression to the entire file, but zipped raw images cannot be operated on directly with regular tools) and often too large to store on FAT formatted media; there is no metadata other than filenames and no internal checksums; missing segments (for example from scratched CD/DVD – can sometimes be overwritten with 0's); and overwritten data is generally unrecoverable (no checksums on small blocks in the file).
- EWF - The primary advantages of EWF are that it includes internal metadata about the imaging process, adds periodic checksums (CRC computed for every 32K block), and supports compression. The primary disadvantage is that EWF is a proprietary format that requires specialized software to read (though it's been thoroughly reverse engineered and documented, and there's a robust set of open-source tools for creating/reading/parsing EWF -

see https://github.com/libyal/libewf/, and one can use that software to generate a raw disk image from the EWF at any point).

How might you use the optional metadata fields in Guymager (3:30)? Would you prefer different labels for those fields? If so, what might they say instead?
Students could suggest a variety of label options such as acquisition number, name of series, name of processing archivist. Note: Euan Cochrane has created files for this kind of customization (https://digitalcontinuity.org/post/183633650358/editing-guymager-user-interface-field-names).

Notice that Guymager has a setting for "split size." (3:15-4:50). What does that do, and why would you want to split a disk image?
Both raw disk images and EWF disk images can be split into multiple files that software then treats as a single disk image. In the case of raw images, the file naming convention is diskimagename.001, diskimagename.002, etc., while the file naming convention for EWF is diskimagename.E01, diskimagename.E02, diskimagename.E03, etc. The primary reason to split a disk image is so that its component parts don't exceed the maximum file or volume size of available storage (e.g. FAT32 maximum file size is 4 Gigabytes).

Guymager has a setting for checksum/hash calculation and validation (4:12-4:46). [Note: It can be helpful to point out that, while the two terms have somewhat different meanings, most people use the terms "cryptographic hash" and "checksum" interchangeably.] What is a checksum/hash?
A given bitstream, fed into an algorithm, will generate a short string of characters (cryptographic hash) that is extremely unlikely to be generated by a different bitstream fed into that same algorithm.

Why might you want to generate and validate checksums?
Checksums serve as essential integrity metadata. You can use checksums to determine whether:

- bits have changed after a transfer - sender and recipient both run the same hash algorithm on the bits and confirm that they match (otherwise resend)

- bits have flipped within a storage environment - store the hash as metadata and periodically run the hash algorithm on the bits to ensure they still match the stored hashes (otherwise, recover good copies from backup or other storage)
- two different files are identical bitstreams - generate hashes for both and see if they match

Does generating a checksum for a file ensure that the file is an authentic record? Why or why not?

One can run the hash algorithm against a given file and see if the resulting hash matches the hash stored in the metadata for that file. If the two hashes match, this is strong evidence for the integrity of the file's bitstream. However, if parties with malicious intent can break into your system to change the files they could also change the hash metadata associated with the files. So you can't ensure that a given bitstream matches the acquired bitstream unless the overall repository system is secure and trustworthy. Checksums also only validate the file as a bitstream (sequence of 1s and 0s). Authenticity is a much broader concept that involves numerous other factors beyond just the integrity of a specific bitstream. In other words, bitstream integrity (as evidenced by checksums) is generally a necessary but not sufficient condition for authenticity.

What's the difference between selecting "Calculate MD5," "Calculate SHA-1," and "Calculate SHA-256"?

- MD5, SHA-1, and SHA-256 are three different hash algorithms, with increasing levels of robustness (note the number of bits for each hash).
- MD5 - Introduced in 1992, MD5 produces a 128-bit hash. The MD stands for "message digest." From a security perspective, MD5 has been "broken" since 2005. Someone with malicious intent can create two different bitstreams that result in the same MD5 hash (i.e. MD5 collisions). This is rarely a concern when MD5 is used for integrity checks on known items (e.g. verifying that a file was transferred correctly to a repository or that files in storage are still intact). It can be a concern if one is relying on a hash as proof of record authenticity; risks can include cases of internal tampering. MD5 is still widely used, because it is fast to calculate and widely supported

- SHA-1 - Introduced in 1995, SHA-1 produces a 160-bit hash. SHA stands for Security Hashing Algorithm. It is much less susceptible to collisions (getting the same hash output from running the algorithm against two different bitstreams) than MD5, but they're still possible. SHA-1 was deprecated by the National Institute for Standards and Technology (NIST) in 2011. In 2017, Google announced that they generated a SHA-1 collision. However, it took them the equivalent of 6,500 years of central processing unit (CPU) time or 110 years of graphic processing unit (GPU) computation (see https://shattered.io/). So a malicious actor with significant computing resources (e.g., government or large corporation) could potentially foil SHA-1, but it's not currently feasible (with today's computers) for individuals to do so easily.
- SHA-256 - Introduced in 2002, SHA-256 belongs to the SHA-2 family of algorithms. It generates a 256-bit hash.

How would you decide between using MD5, SHA-1, or SHA-256 or a combination of them?
It is common practice now to generate at least two different hashes for each file. Many institutions use an MD5 for doing routine integrity checks, because the calculation is much faster than for SHA-1 or SHA-2 (including SHA-256). They can then use a SHA hash for cases when it's important to ensure that a file is what it purports to be, i.e. no one has generated a file that's composed of a different bitstream but results in the same hash output (called a collision). An added benefit of storing two hashes is that generating a different file that has the same hash output using two different algorithms (MD5 and SHA) is significantly harder than doing so for just a single algorithm.

Guymager provides various types of information in the .info file (in this case image1.info). (5:54-7:34) How would you characterize the types of information you can find in the .info file?
The info file includes information about the software and parameters used for the acquisition, about the device captured, and hashes of the disk image (see also https://youtu.be/mqHx7HutQLo?t=545). In other words, it includes a lot of technical and provenance metadata.

Which information from the .info file would want to retain over time? Why? Who would you see using the information, and how would you see them using it?

Students could answer this in many ways, depending on which elements of the file they choose. For example, one might want to consult the technical metadata about the drive which otherwise would no longer be available once the imaged device is gone or no longer working.

The .info file has a MD5 hash and verified MD5 hash for the disk image (6:47-7:07). A hash is a specific output based on a specific input. What was the input to calculate this MD5 hash?

The input to the hash function is the raw disk image data, which is the bitstream generated by reading each storage sector from the drive and writing the results to a file.

If I were to run a checksum tool against the file called image1.E01 (5:31), would I get the same result MD5 hash as the one stored in the .info file? Why or why not?

The two hashes would not be the same. The MD5 hash stored in the metadata is based on the raw disk image data. The entire .E01 file includes added metadata, periodic integrity checks and may be compressed, so it is not the same bitstream as the raw disk image data, and an MD5 hash based on the entire .E01 file will be different.

Tools and Resources Mentioned in This Document

Guymager:

- https://confluence.educopia.org/display/BC/Imaging+with+Guymager
- https://guymager.sourceforge.io/

Chapter 5. Virtual Storage. (n.d.). Retrieved November 3, 2021, from https://www.virtualbox.org/manual/ch05.html#vdidetails

Cochrane, E. (2019, March 23). Editing Guymager User Interface Field Names. *Digital Continuity Blog*. https://digitalcontinuity.org/post/183633650358/editing-guymager-user-interface-field-names

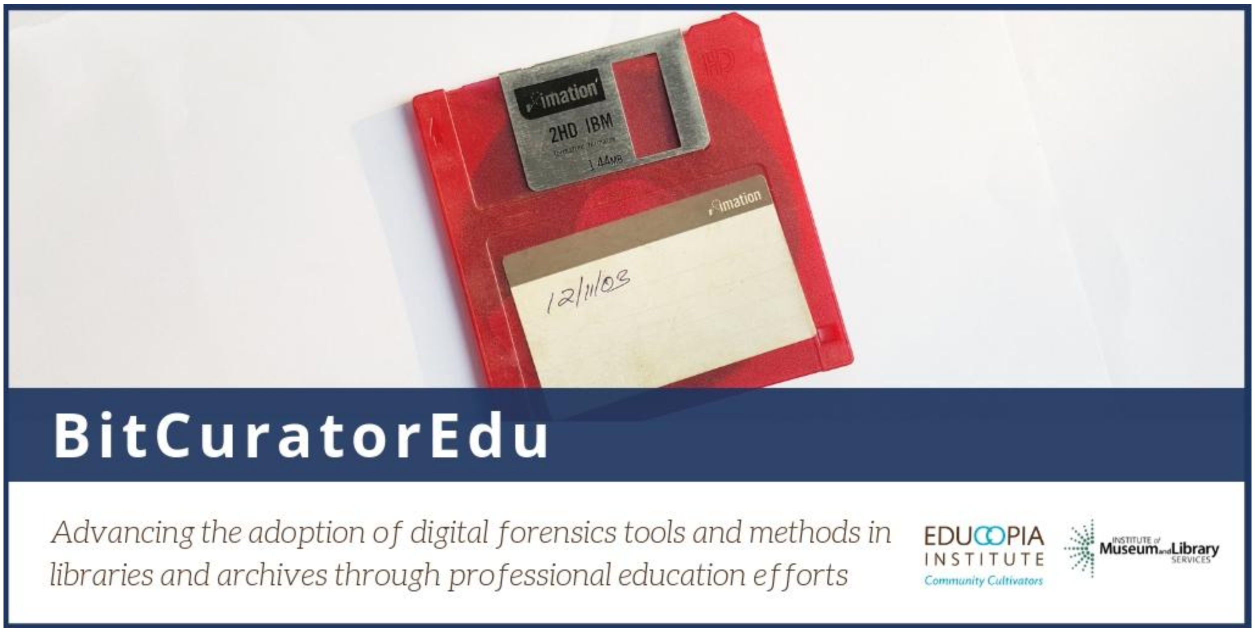
DFIRScience. (2016, October 3). *Forensic Acquisition in Linux—Guymager*. https://www.youtube.com/watch?v=mqHx7HutQLo

Garfinkel, S., & Cox, D. (2009). *Finding and Archiving the Internet Footprint*. NAVAL POSTGRADUATE SCHOOL MONTEREY CA. https://apps.dtic.mil/docs/citations/ADA549352

Loop device. (2021). In *Wikipedia*. https://en.wikipedia.org/w/index.php?title=Loop_device&oldid=1012253622

SHAttered. (n.d.). Retrieved November 3, 2021, from https://shattered.io/

BitCuratorEdu Learning Object


imation
2HD IBM
1.44MB
imation
12/11/03
BitCuratorEdu
Advancing the adoption of digital forensics tools and methods in libraries and archives through professional education efforts
EDUCOPIA INSTITUTE
Community Cultivators
Institute of Museum and Library Services

Most resources from the BitCuratorEdu project are intentionally left with basic formatting and without project branding. We encourage educators, practitioners, and students to adapt these materials as much as needed and share them widely.

The BitCuratorEdu project is a three-year effort (2018-2021) funded by the Institute of Museum and Library Services (IMLS) to study and advance the adoption of digital forensics tools and methods in libraries and archives through professional education efforts. This project is a partnership between Educopia Institute and the School of Information and Library Science at the University of North Carolina at Chapel Hill, along with the Council of State Archivists (CoSA) and several Masters-level programs in library and information science.