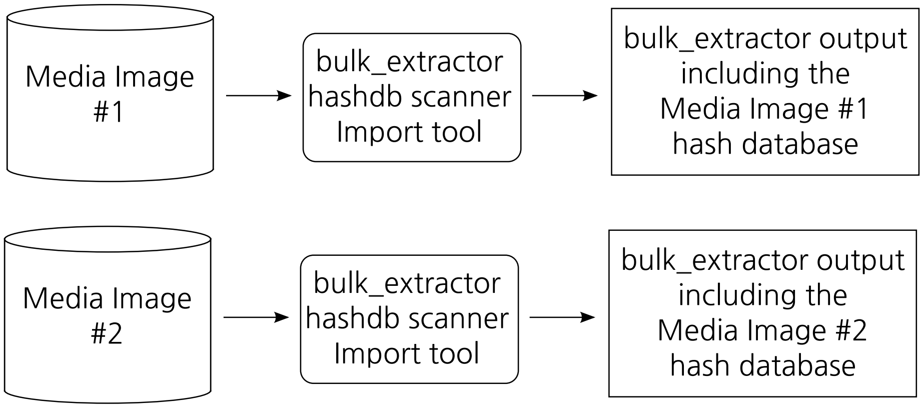
Demo: Finding Similarities between Media Images
Using *bulk_extractor* and *hashdb*

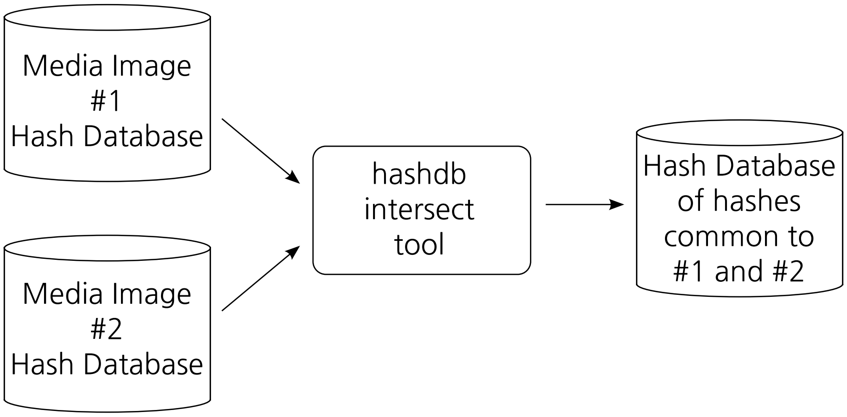
In this demo, we find similarities between media images by finding block hashes that are common between them. Here are the steps:

1. Generate a hash database of block hashes from media image 1.
2. Generate a hash database of block hashes from media image 2.
3. Obtain common block hashes by taking the intersection of these two databases.

Here is the workflow:

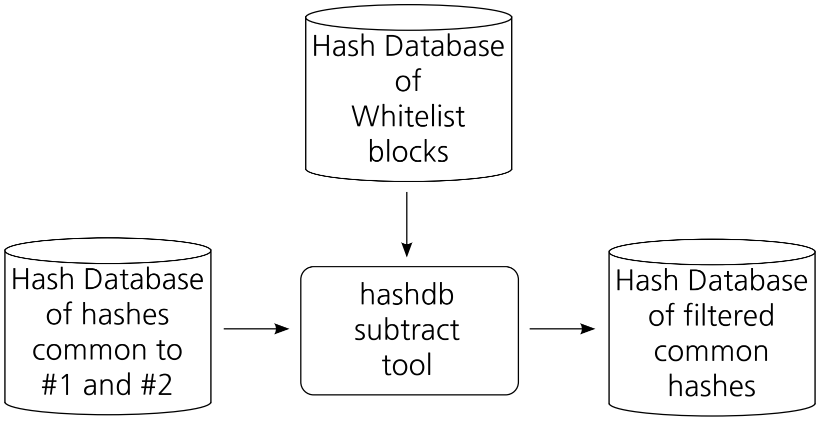

Media Image #1
bulk_extractor hashdb scanner Import tool
bulk_extractor output including the Media Image #1 hash database
Media Image #2
bulk_extractor hashdb scanner Import tool
bulk_extractor output including the Media Image #2 hash database

Run *bulk_extractor* with the *hashdb* `import` option selected to import media images into hash databases.


Media Image #1 Hash Database
Media Image #2 Hash Database
hashdb intersect tool
Hash Database of hashes common to #1 and #2

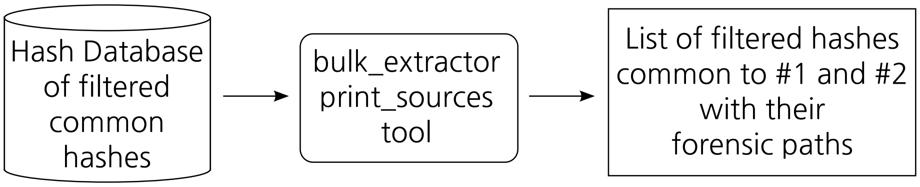
Run the *hashdb* `intersect` command to generate the database of common hashes.

Media Images contain many false positives because they contain null blocks, operating system files, common header blocks, etc. To remove these false positives from our hash dtabase, we subtract out hashes if they are are already known:


Hash Database of Whitelist blocks
Hash Database of hashes common to #1 and #2
hashdb subtract tool
Hash Database of filtered common hashes

Run the *hashdb* `subtract` command to create a database without specified hashes.

Now view the list of forensic paths common to both media images:


Hash Database of filtered common hashes
bulk_extractor print_sources tool
List of filtered hashes common to #1 and #2 with their forensic paths

Run the *hashdb* `get_sources` command
to list common hashes and where they were sourced from.

Here are the steps to perform this demo:

1. Download and install *hashdb* from `http://digitalcorpora.org/downloads/hashdb` as described at `https://github.com/simsong/hashdb/wiki/Installing-hashdb`.
2. Download and install *bulk_extractor* compiled with *hashdb* from `http://digitalcorpora.org/downloads/hashdb` as described at `https://github.com/simsong/hashdb/wiki/Installing-hashdb`.
3. Identify two media images to compare. I'll call them `media_image_1` and `media_image_2`. Note that images are available at `http://digitalcorpora.org/corpora/disk-images`. For example `http://digitalcorpora.org/corp/nps/scenarios/2009-m57-patents/drives-redacted/jo-favorites-usb-2009-12-11.E01` and `http://digitalcorpora` `org/corp/nps/scenarios/2009-m57-patents/drives-redacted/jo-work-usb-200` `E01`.
4. Import media image 1:
   ```
   $ bulk_extractor -e hashdb -o outdir1 \
     -S hashdb_mode=import media_image_1
   ```
5. Import media image 2:
   ```
   $ bulk_extractor -e hashdb -o outdir2 \
     -S hashdb_mode=import media_image_2
   ```

6. Create the intersection of media image 1 and 2 to obtain a database of hashes that are common between them:

   ```
   $ hashdb create intersection.hdb
   $ hashdb intersect media_image1/hashdb.hdb media_image2/hashdb.hdb \
            intersection.hdb
   ```

Now database `intersection.hdb` contains common hashes, but we want to know what is in it. Here are some approaches for further analysis:

- There are "false positives" from system files. Which hashes are actually interesting user data? Subtract false positives from a database of files known to be not interesting, such as system files, so that the remaining hashes are pertinent data common between the media images:

  ```
  $ hashdb create intersection2.hdb
  $ hashdb subtract intersection.hdb false_positives.hdb \
           intersection2.hdb
  ```

- What hashes match previously encountered data? Intersect the media image intersection database with a database of known interesting data, and print out the list of files from which the known hashes were sourced:

  ```
  $ hashdb create recognized_hashes.hdb
  $ hashdb intersect intersection.hdb known_hashes.hdb \
           recognized_hashes.hdb
  $ hashdb get_sources recognized_hashes.hdb > output.txt
  ```

Note that these approaches for further analysis require databaes of previously encountered files. For a demo on preparing a block hash database, please see `http://digitalcorpora.org/downloads/hashdb/demo/create_hdb_demo.pdf`. To create a database of "false positives", import hashes from some media images or from the NSRL.

Note that this demo uses the default hash block size of 4096 bytes and the default sector size of 512 bytes. Please see *bulk_extractor* usage for options.