Maria Valencia

CSC 153

Lab 9

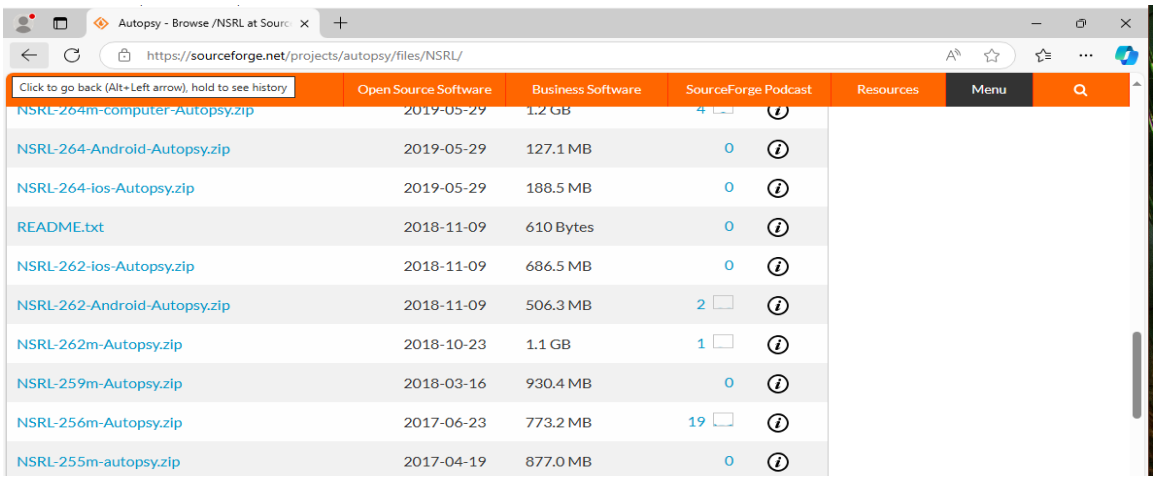
Lab 9: Analysis and Validation

Task 1: Autopsy Hashsets

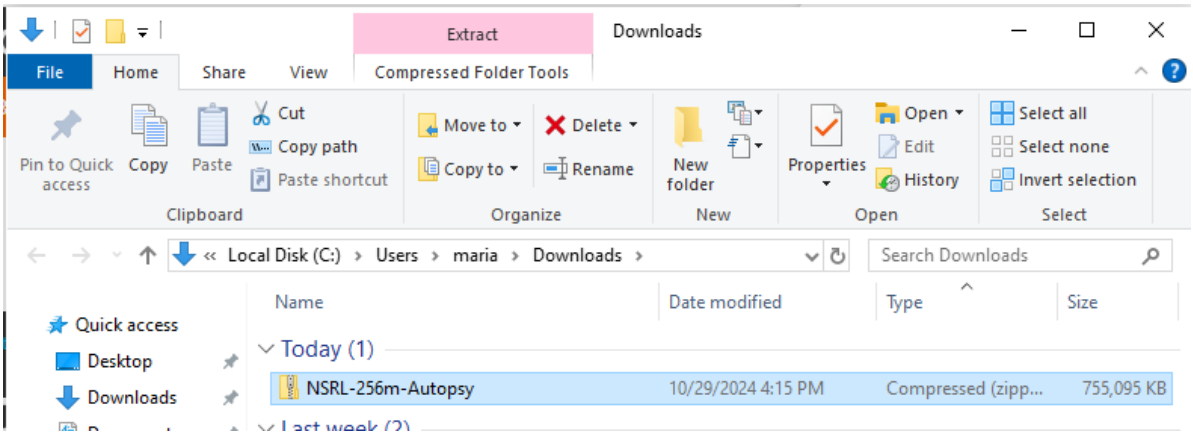
In this lab, I integrated the NSRL into Autopsy and used it in a case from my Windows VM.

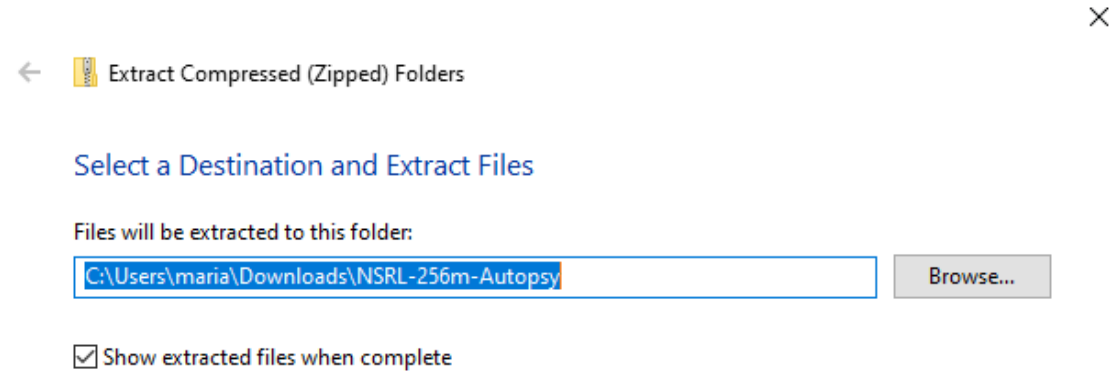
Step 1: Setup NSRL in Autopsy

From my windows vm, I downloaded the NSRL from https://sourceforge.net/projects/autopsy/files/NSRL/ and downloaded the "NRSL-256m-Autopsy.zip" file only.

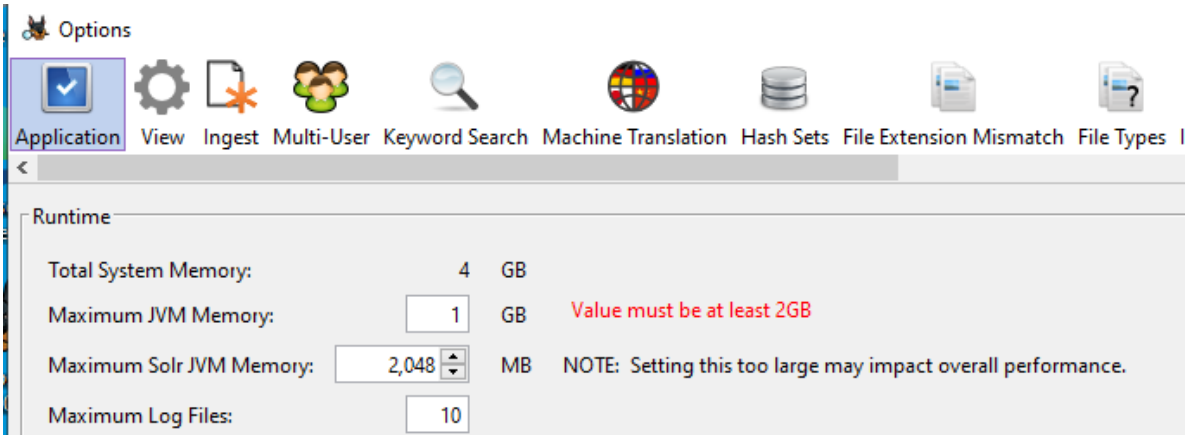


Once it was downloaded, I unzipped the file contents by selecting it in my downloads folder and extracting all files.

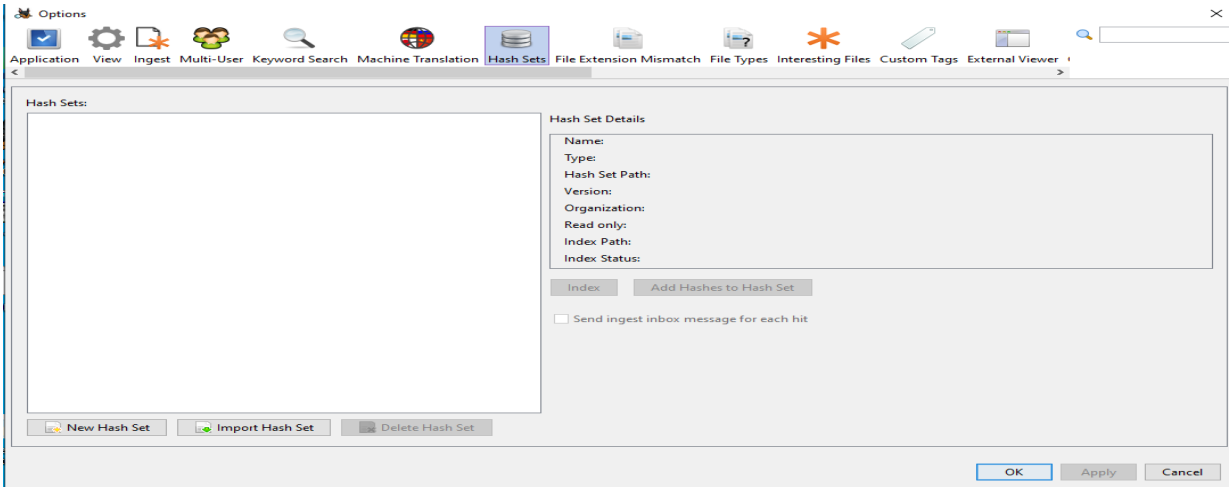




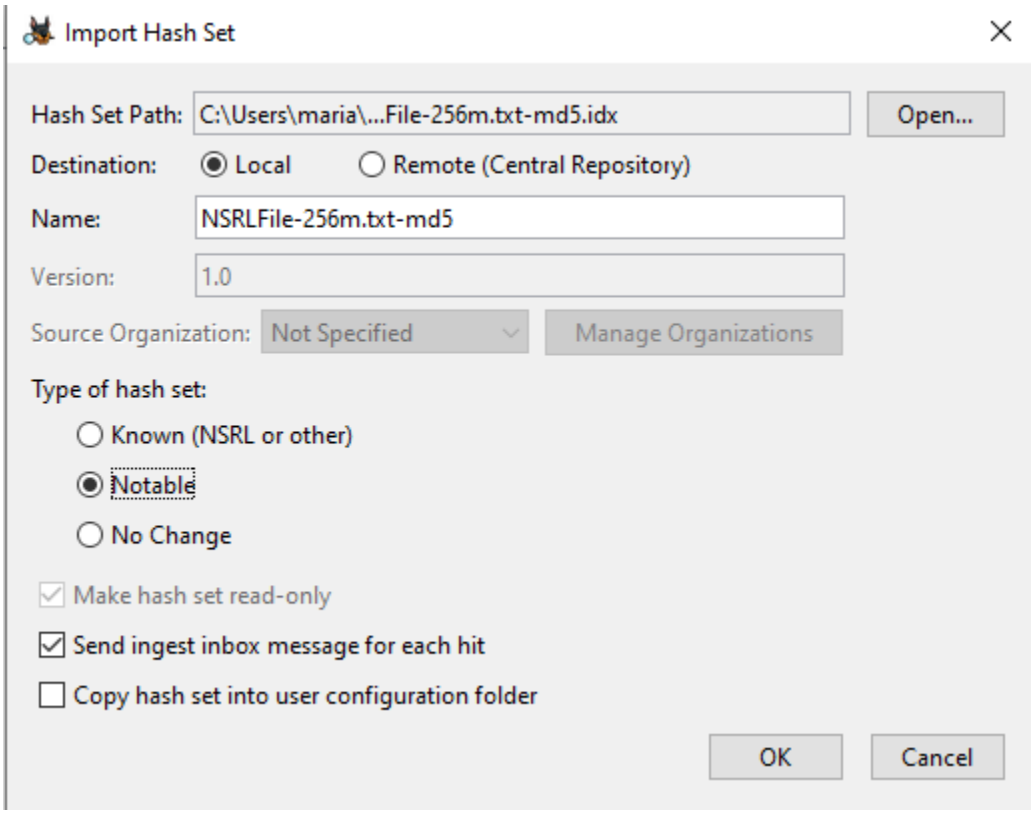
I launched autopsy and closed the welcome window. Then navigated to Tools and chose Options from the menu. While under the Application tab, I increased the Maximum JVM memory to 2GB.



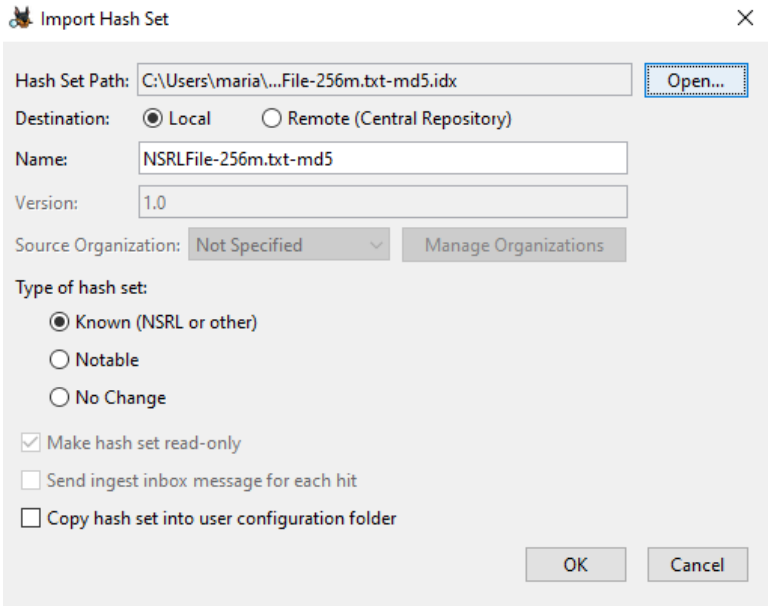
In the options menu, I selected Hash Sets (top icon bar) and pressed the import hashsets button.

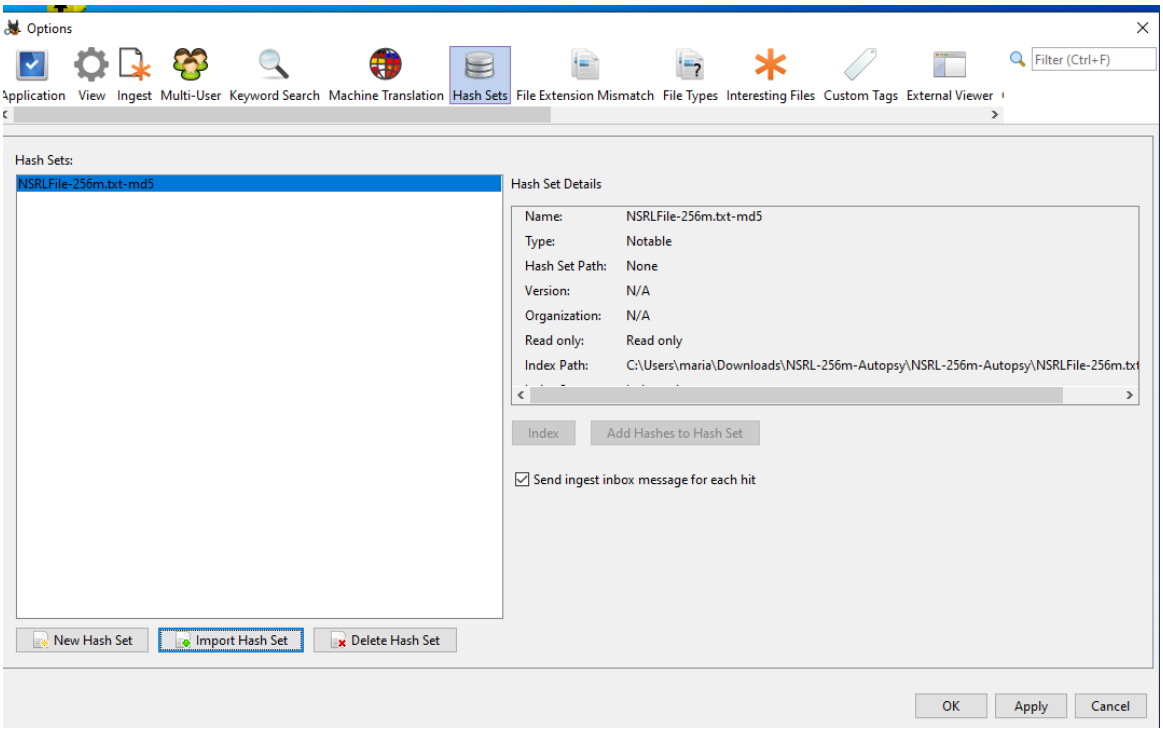


With the Import Hash Database window launched, I selected the extract “NSRLFile-256m.txt-md5.idx” file, selected Notable for the type of hash set and pressed ok.



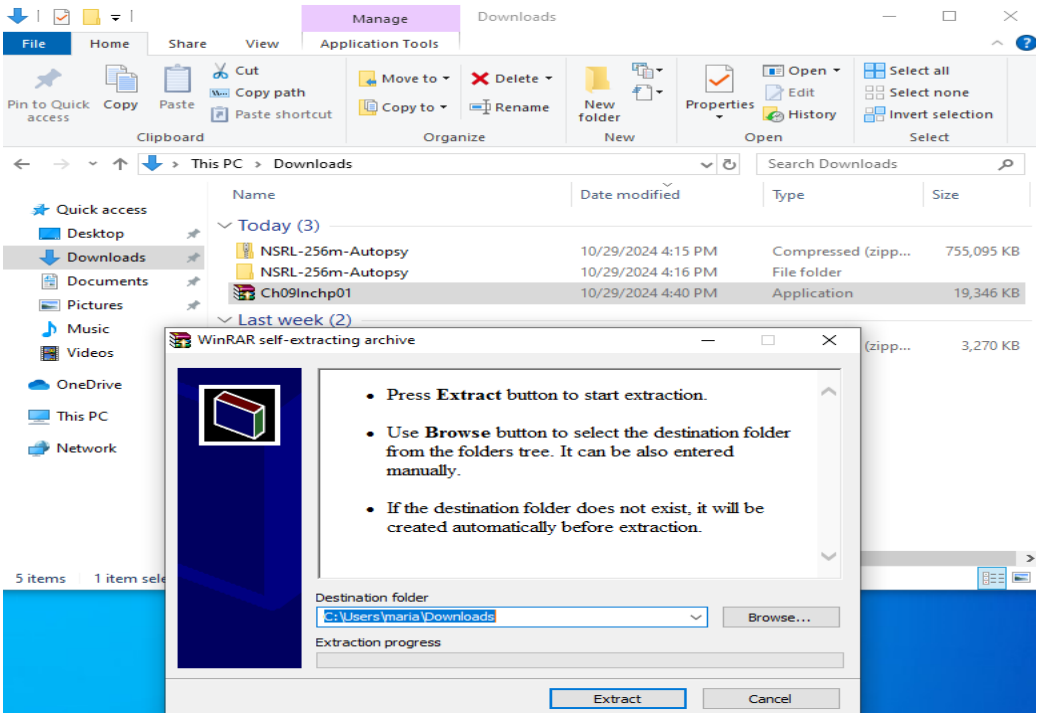
After the NRSL file had been entered, I pressed Apply and then OK to complete the import.



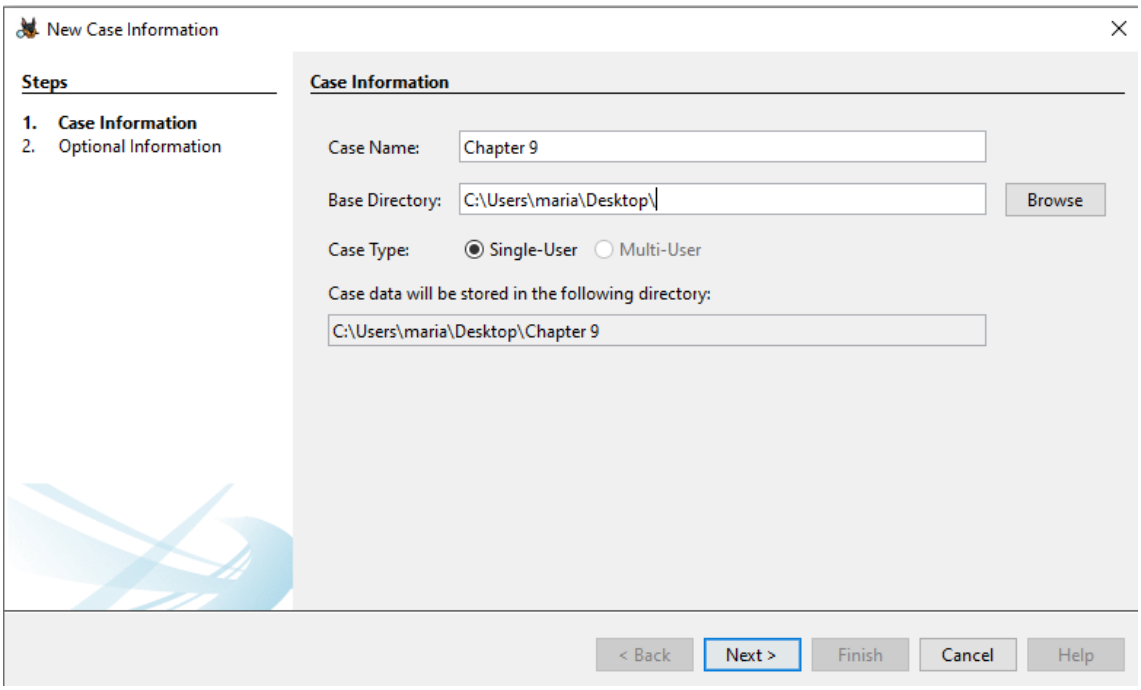


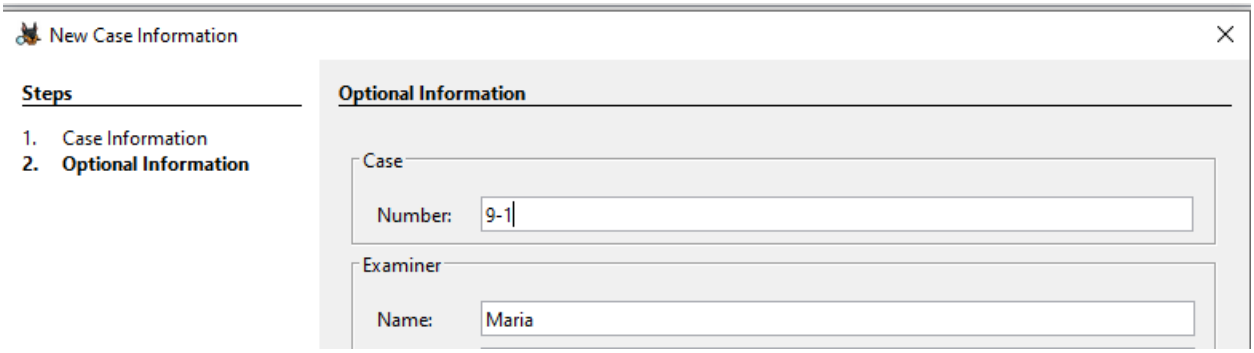
Step 2: Create a Case

I downloaded the “Ch09Inchp01.exe” file and copied it to my windows vm. Once it was on the VM, I double clicked the exe to extract the image.

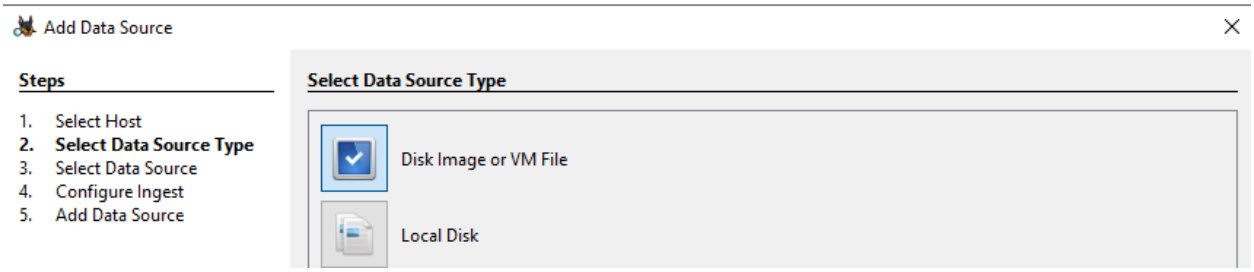


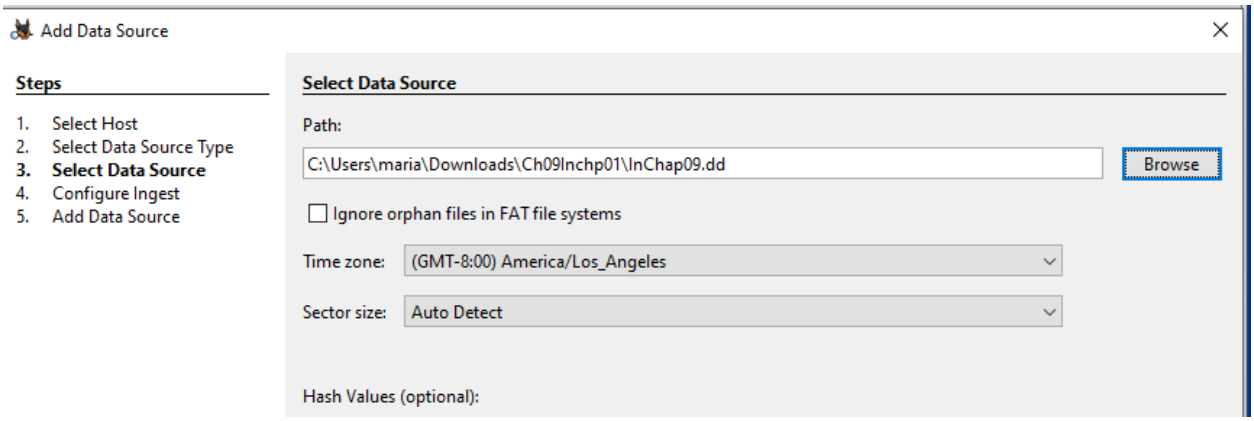
I created a new case in Autopsy with the name “Chapter 9”, case number “9-1” and my name as the examiner.



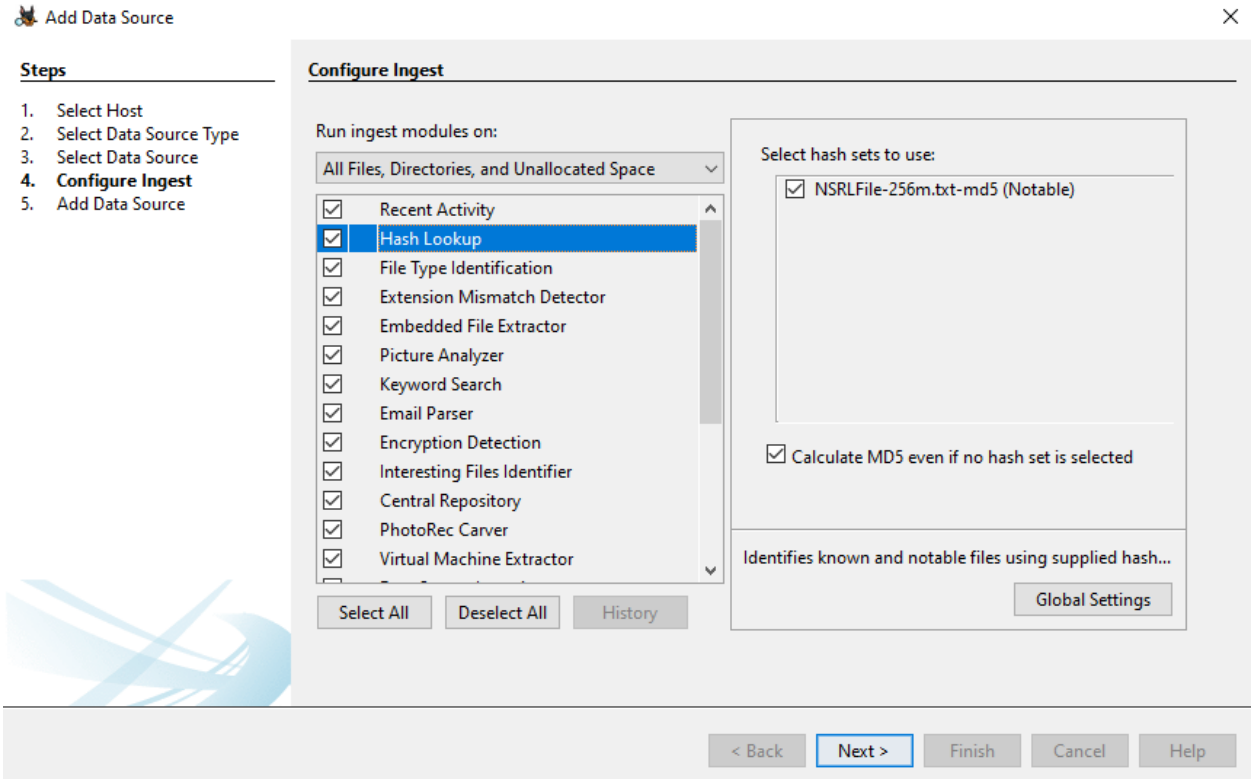


I added the extracted “InChp09.dd” file as the data source for my case as a Disk image or VM.



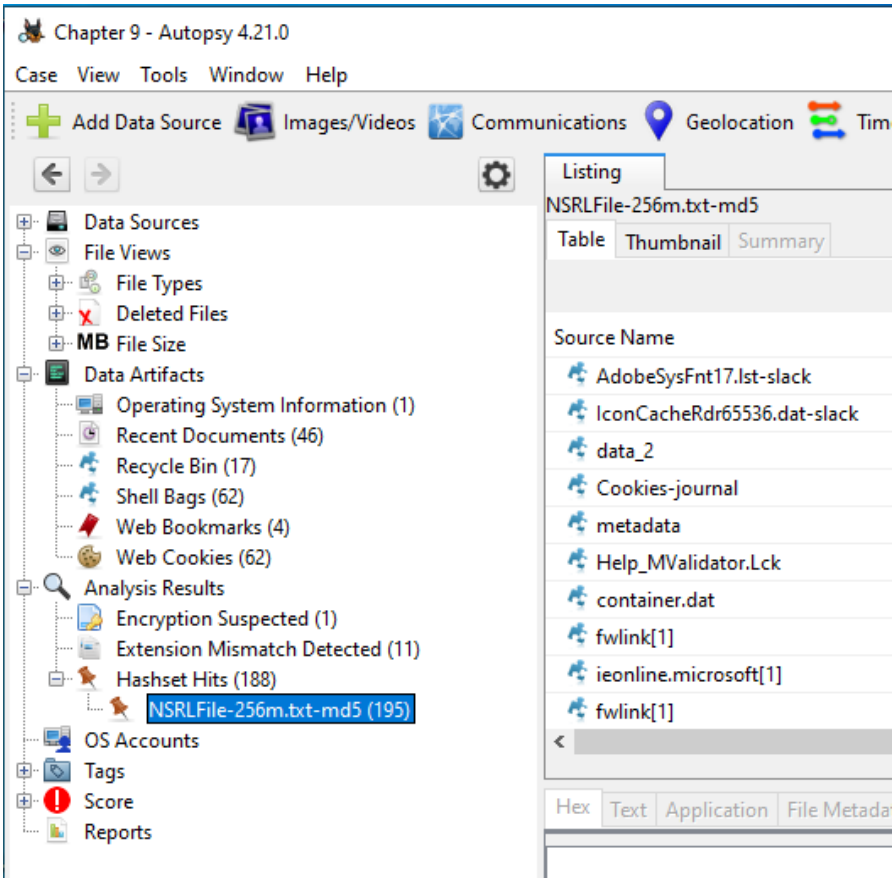


On the configure ingest modules steps of the Add Data Source, I observed the loaded NSRL database under the Hash lookup ingest module. I selected Next and Finish.

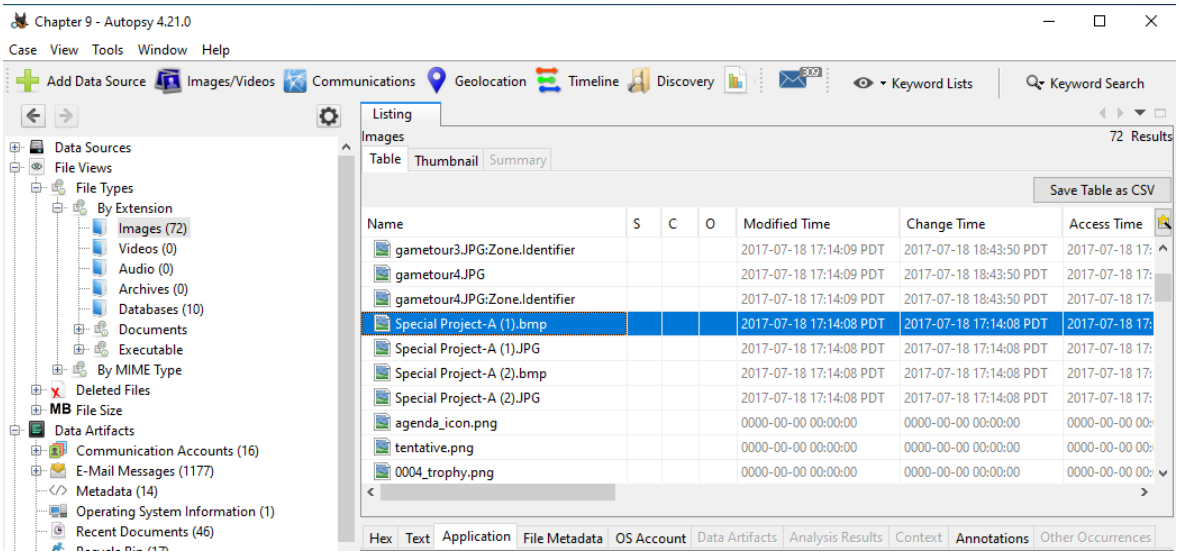


Step 3: Hashset Tracking

With the case created and data source analyzed in autopsy, i selected Results and Hashset Hits in the left pane tree. I observed that Autopsy identified hash hits for this image.

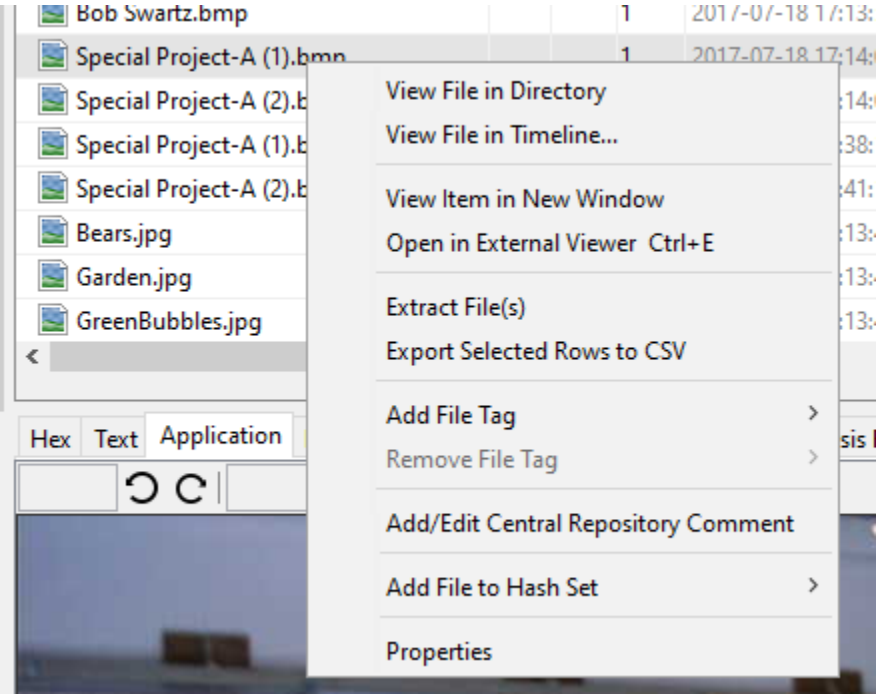


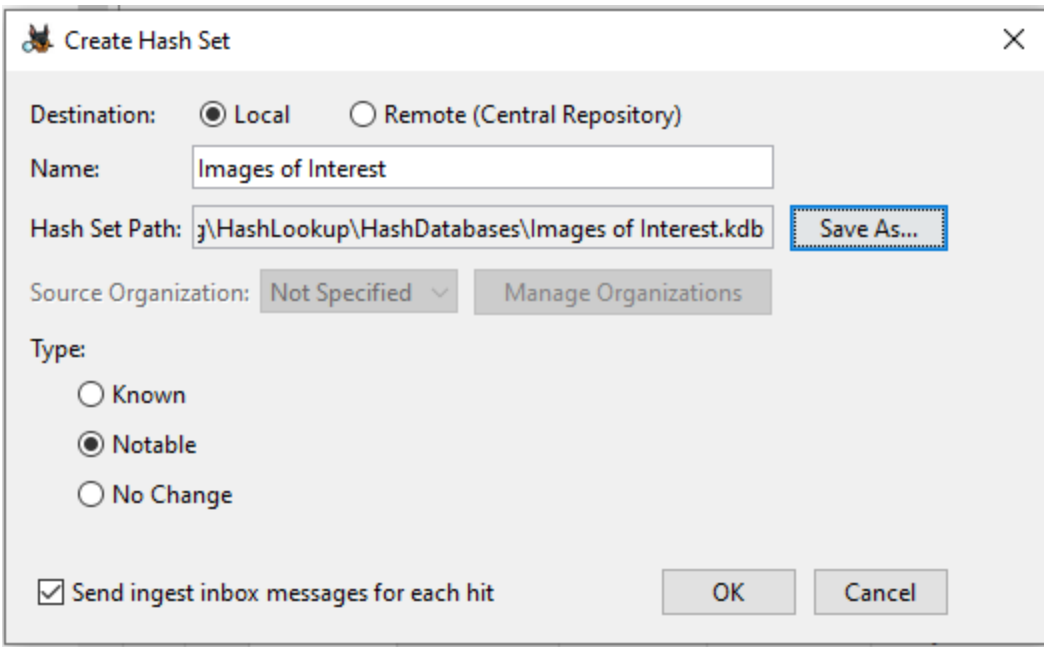
On the left navigation pane, I expanded file views, file types, by extension, and select images. I scrolled to the “Special Project-A" image files in the directory listing tab. I assumed these files are known to be of interest to our investigation.



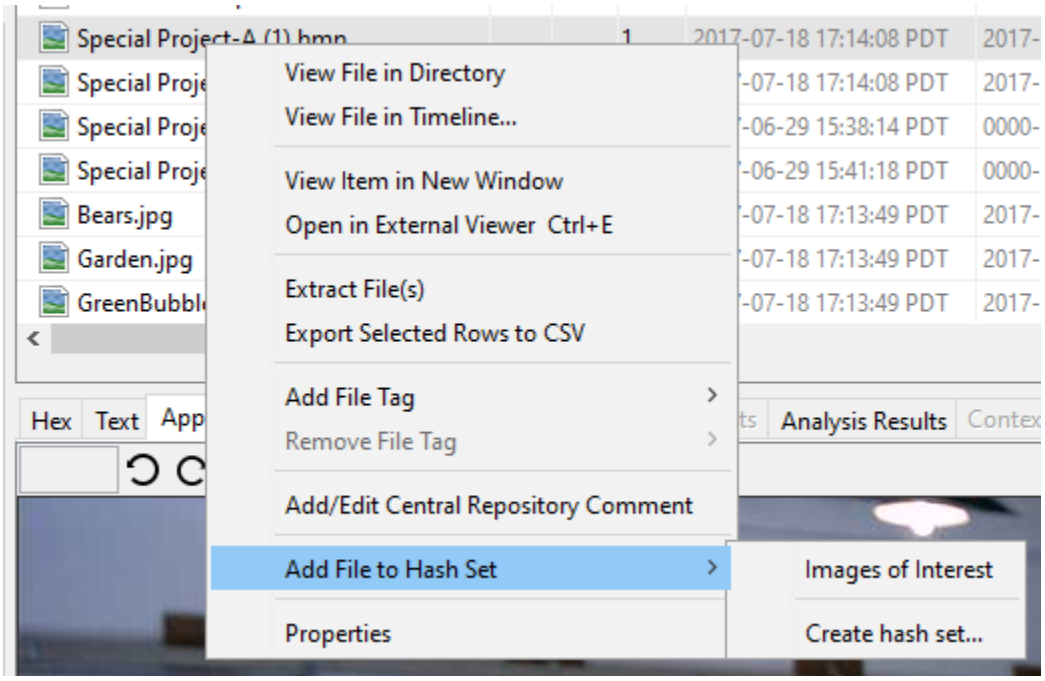
I right Clicked the Special project-A(1).bmp file and selected create hash set under the add file to hash set menu.

I named the Hash set “Images of interest” and the type as notable. I selected save as… next to the database and hit save to leave as the default location.

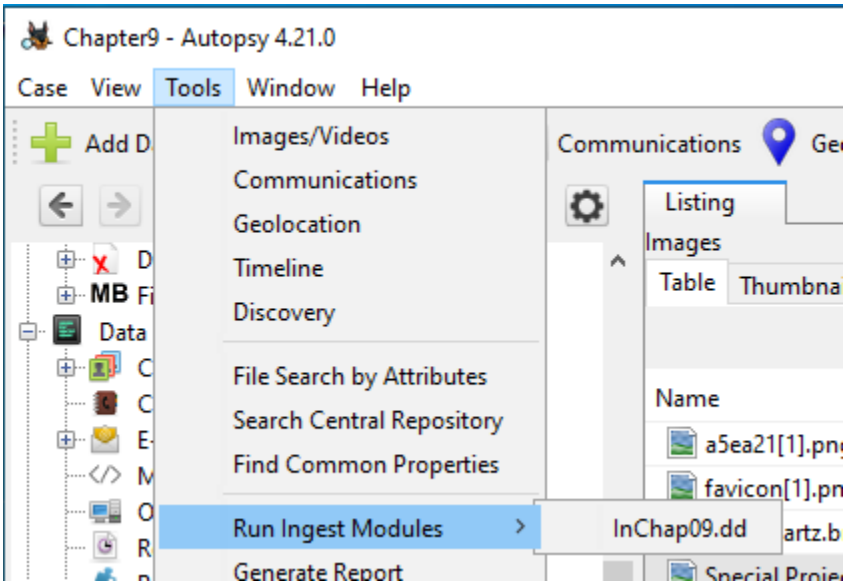




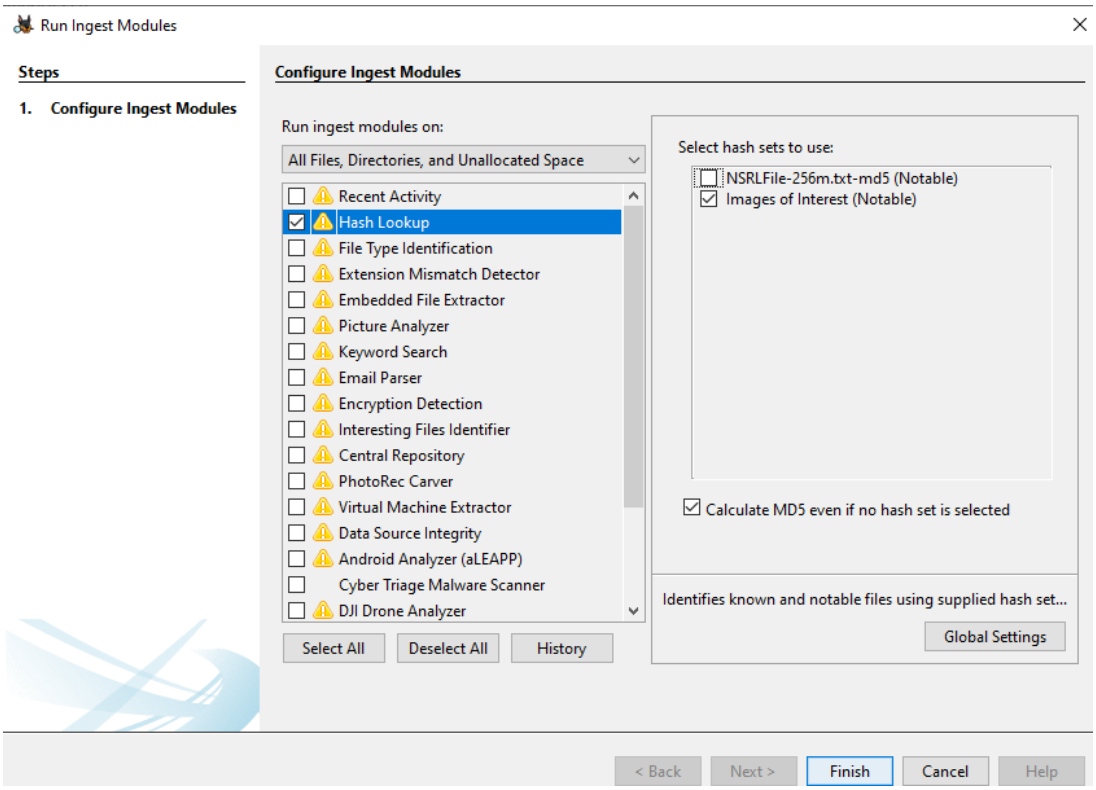
With the database created, I right-clicked the “Special Project-A(1).bmp file again and selected images of interest under the add file to hash database menu.



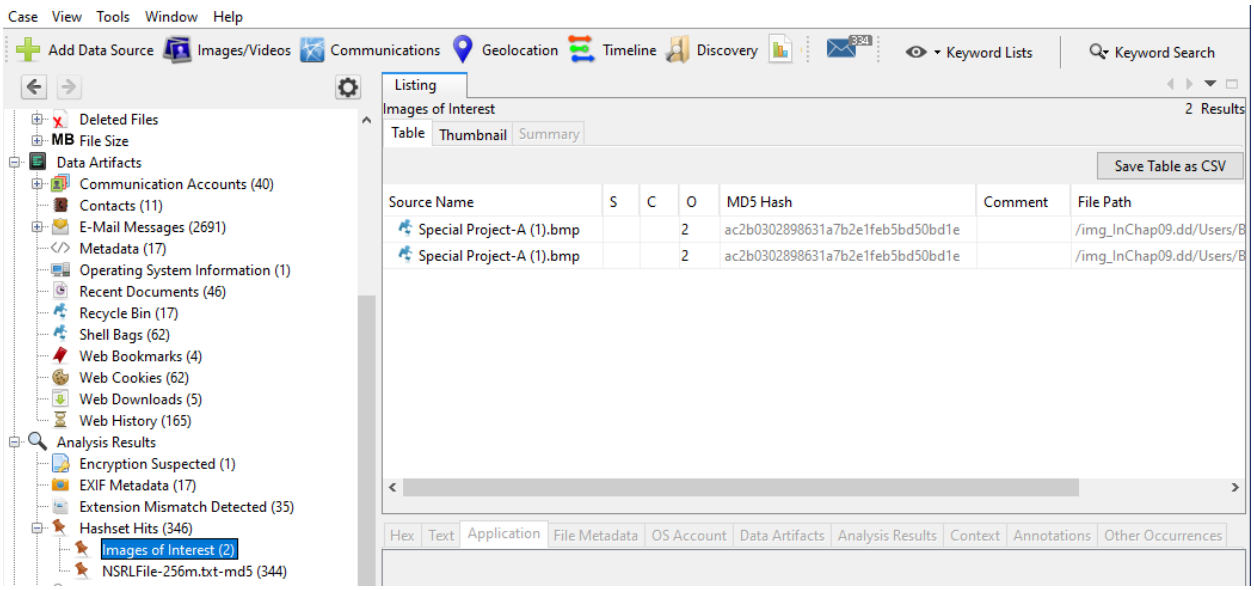
With the file added to the images of interest hash database, I selected tools, run ingest modules and selected the InChp09.dd image.



I deselected all the modules except for hash lookup. I selected the Hash lookup module and observed our ‘Images of interest” set listed. I pressed start to start the re-ingestion.



After the module reruns, I observed the hashset hits under results in the left navigation tree includes the identified images of interest based on it md5 hash!



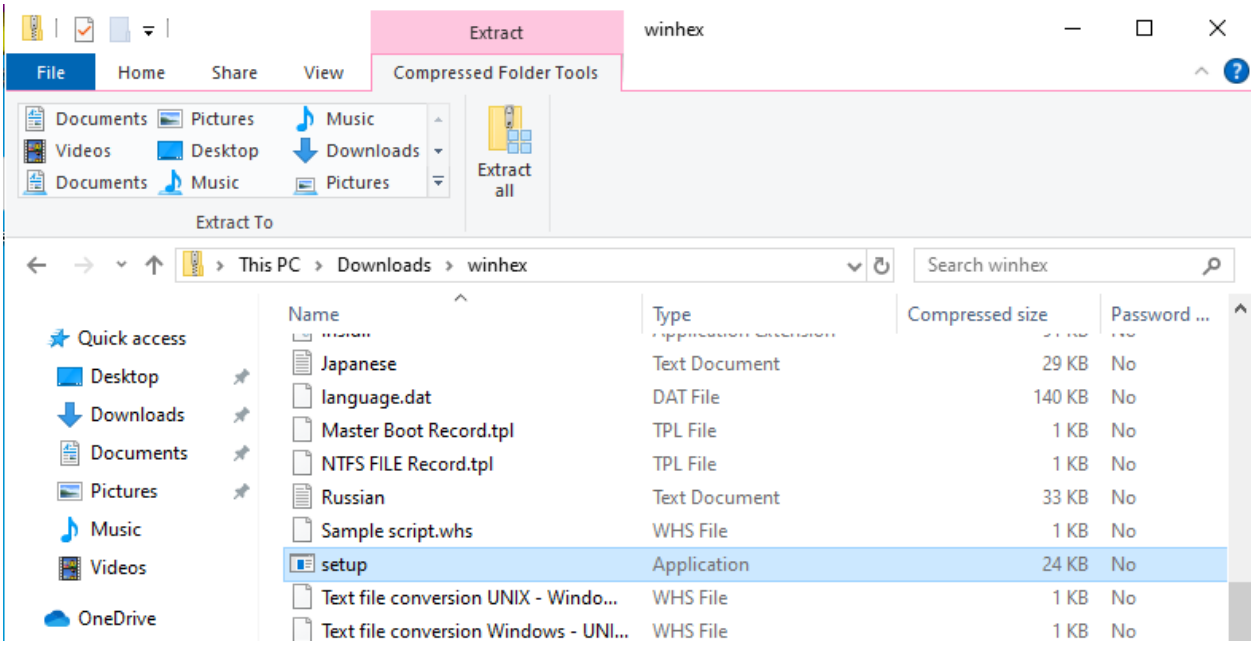
Task 2: Hashing with WinHex

This task explores hash generation with WinHex on my Windows VM.

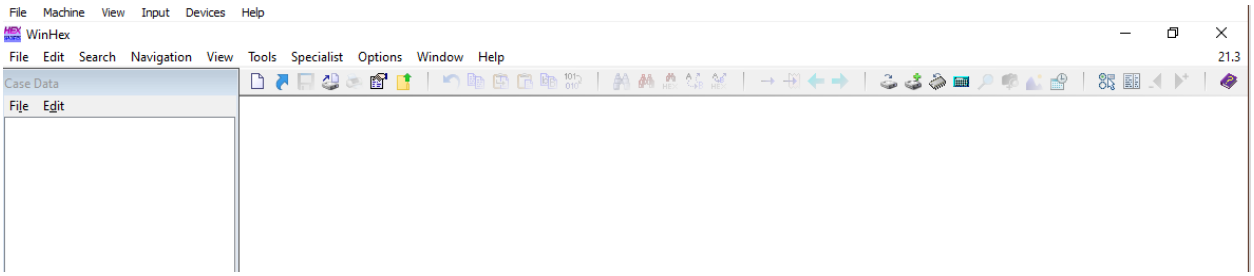
Step 1: Install WinHex

I navigated to https://www.x-ways.net/winhex/ and downloaded WinHex.

Navigated to my downloads folder and extracted the zip file and began installation.

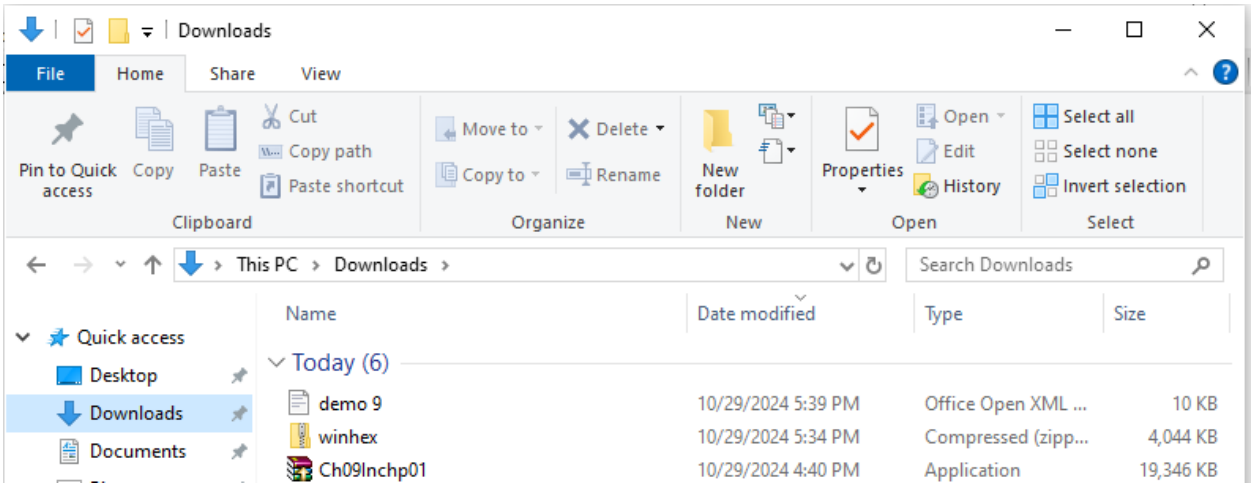


I accepted UAC prompt and used the default settings until installed. I pressed OK and winhex opened.

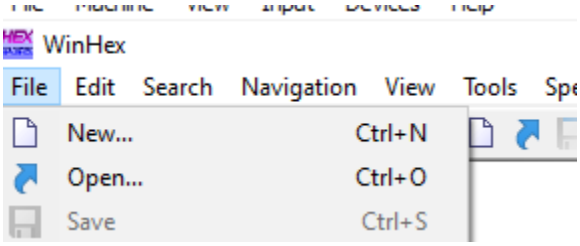


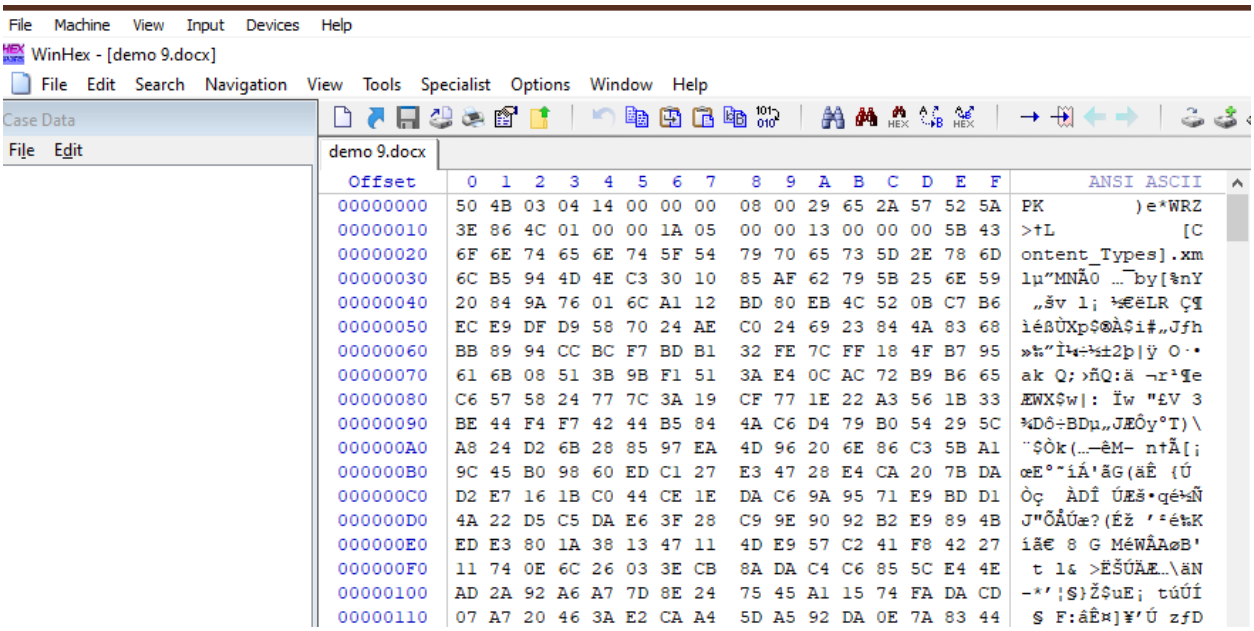
Step 2: Hash a word file

I copied the “demo 9.docx” file onto the VM.

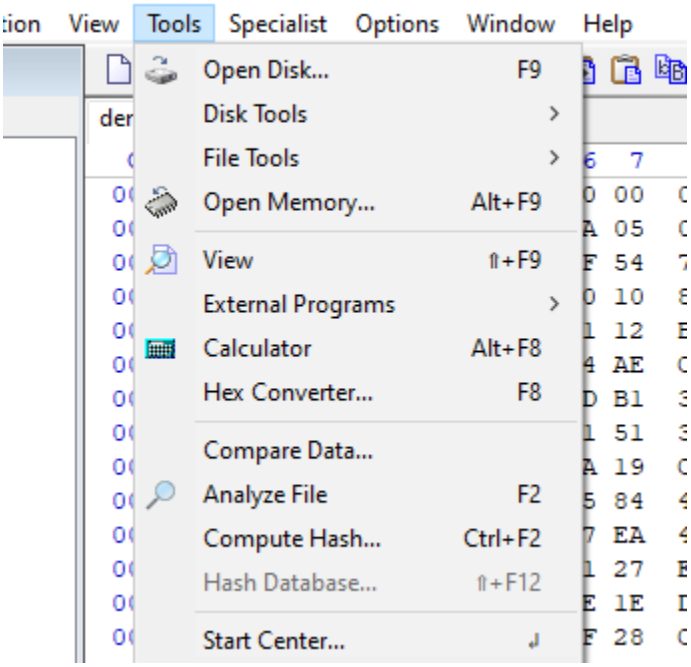


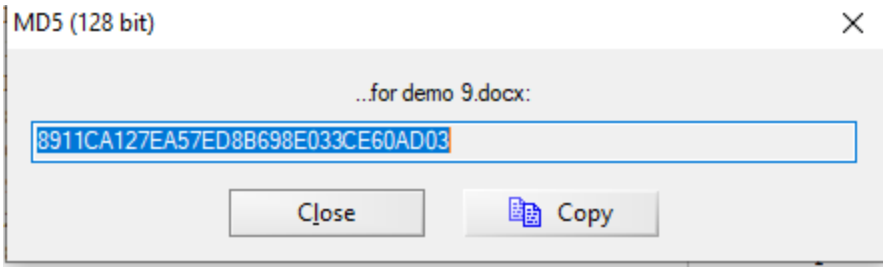
With WinHex running, I opened the demo 0 docx file from the File and Open menu bar.





I computed the hash of the file using the Tools and compute hash tool from the menu. I selected MD5 and OK.





Task 3: Bit-Shifting with WinHex

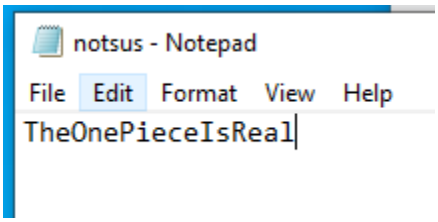
In this task I used Bit-Shifting technique to obfuscate a file's contents using WinHex on my windows VM.

Step 1: Install WinHex

Already did in previous task.

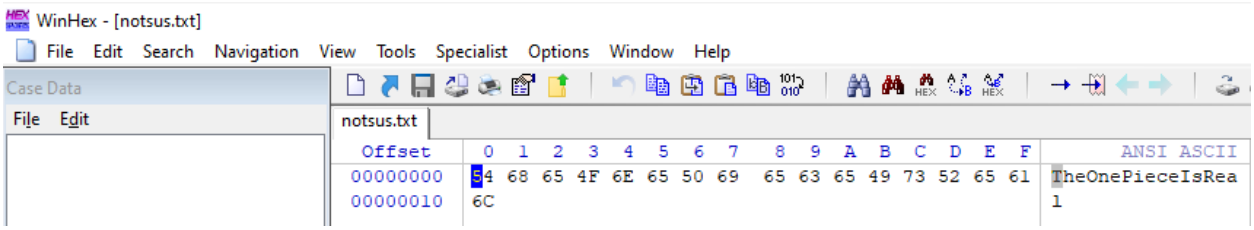
Step 2: Create a Secret File

I created a file "notsus" with a short phrase.

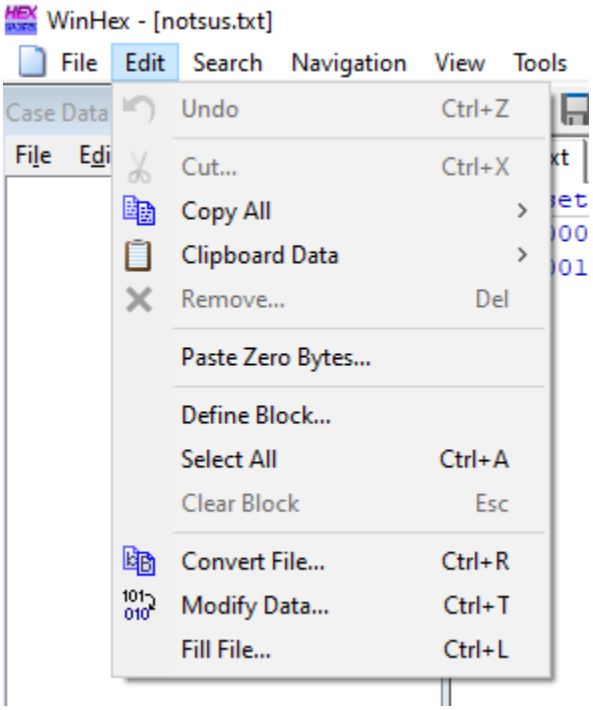


Step 3: Bit-Shift Secret File

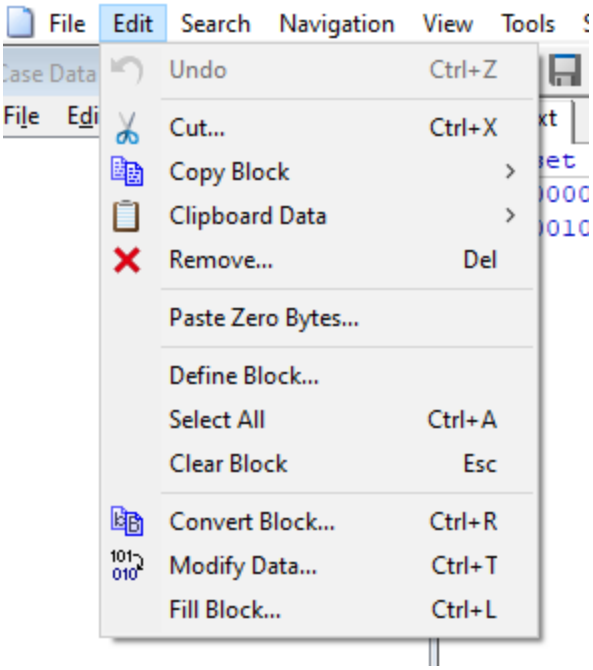
I launched WinHex and opened the text file I created.

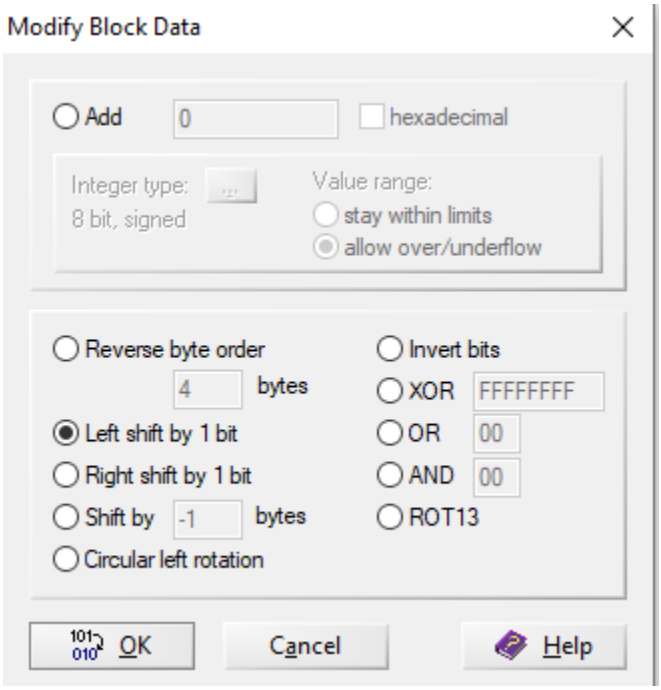


I Highlighted all the data in the file by selecting Edit and select all from the menu.

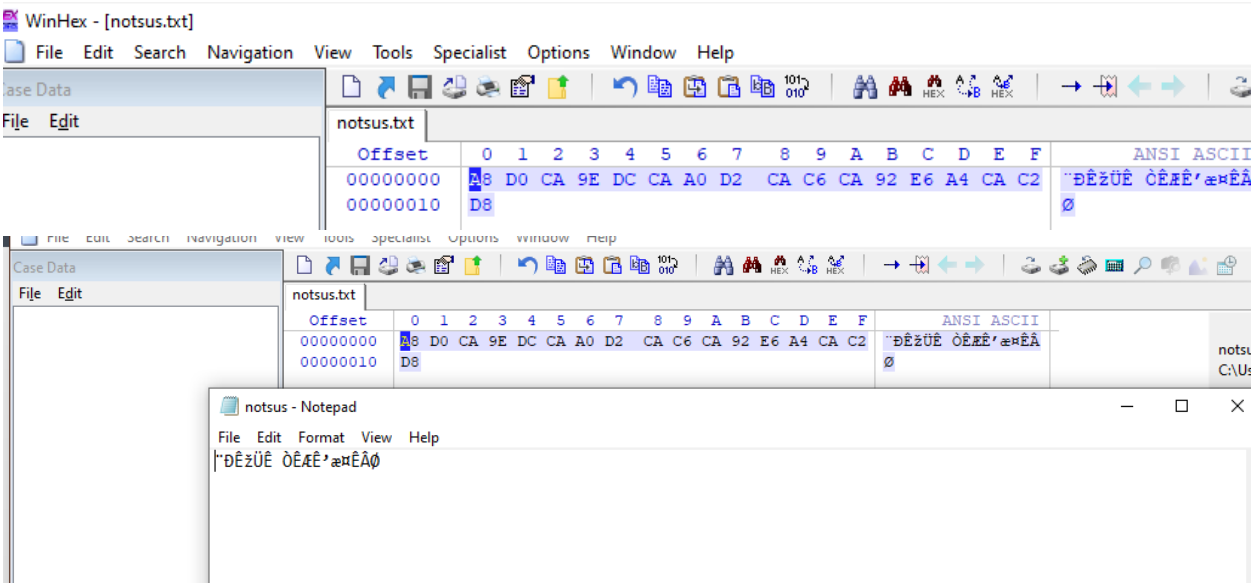


With the contents selected, I modified the data by electing edit and modify data from the menu. In the modify block data, I selected “left Shift by 1 bit” option and pressed ok.



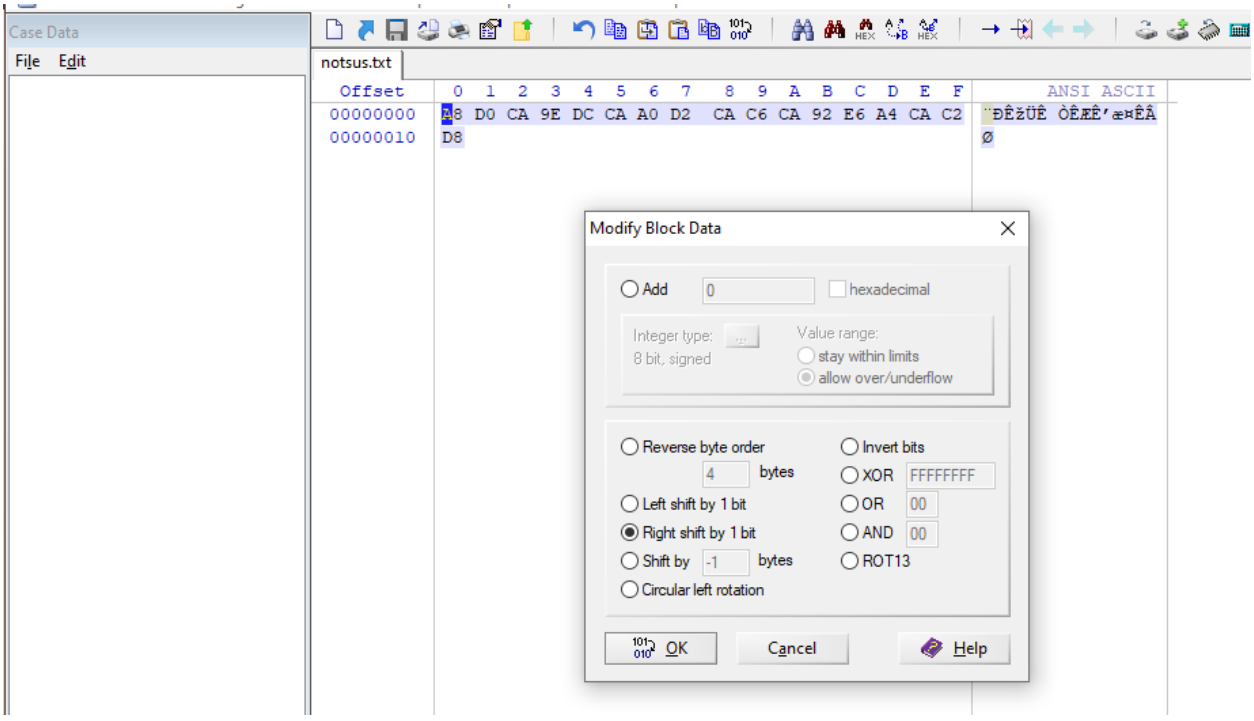


I observed that all the data has now been altered and the ascii representation is no longer readable. I saved the file and opened it in notepad. I observed that the text is not legible

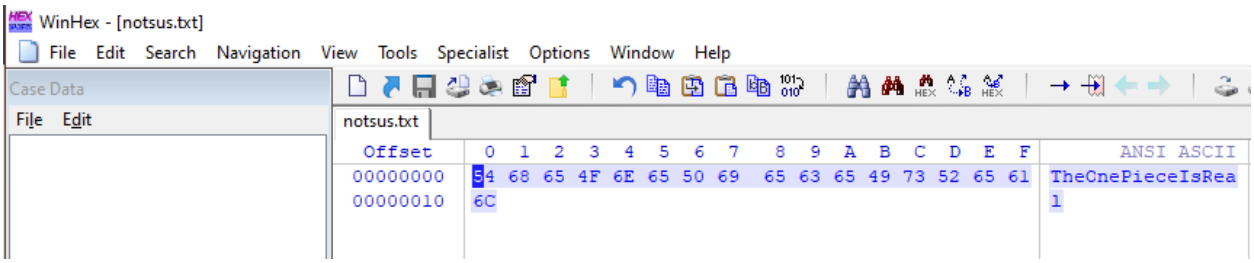


Step 4: Recover Bit-Shift File

With WinHex launched and bit-shift left text file open, shifts bits right to recover the original data. I selected all data in the file, launch Modify Data in the edit menu, select right shift by 1 bit and OK.



My data has been recovered!



The one piece is truly real. Haha this was a fun lab professor, thanks!