

**Christopher HAWKINS**

**09002003**

**MSc Computer Science (Cyber Security)**

**School of Digital, Technologies and Arts**

**Supervisor: Dr Mohamed HASSAN**

**MSc Dissertation Project (JAF)**

**DEVELOPMENT DIARY & CHANGELOG**



For consistency of testing, only the authors machines were used in the development and testing of this tool – the VM that was used was identical:

Specs:

DESKTOP

Windows 10 Home Edition

Intel Core i7-9700k @ 3.6GHz

48GB RAM

4.5TB Storage

NVIDIGA GeForce 1660Ti GPU

LAPTOP

VM

Parrot Security based on 64-bit Debian Linux

Base Memory: 10GB

Video Memroy 128MB

200GB Virtual HDD

**PRE-240521 – V 0.0.0**

This was the preparatory time and included creating the script, downloading sample files for testing, and putting in basic variables and arguments and ensuring that the basic functionality of the *volatility* tool can be run through the script.

**240521 – V 0.1**

Meeting with DM arranged for 28/05/21 to discuss tool and output. Decided questionnaire would be completed to validate output.

Read through script and came up with additional functionality. Data assurance added via sha256sum commands, tested and functional.

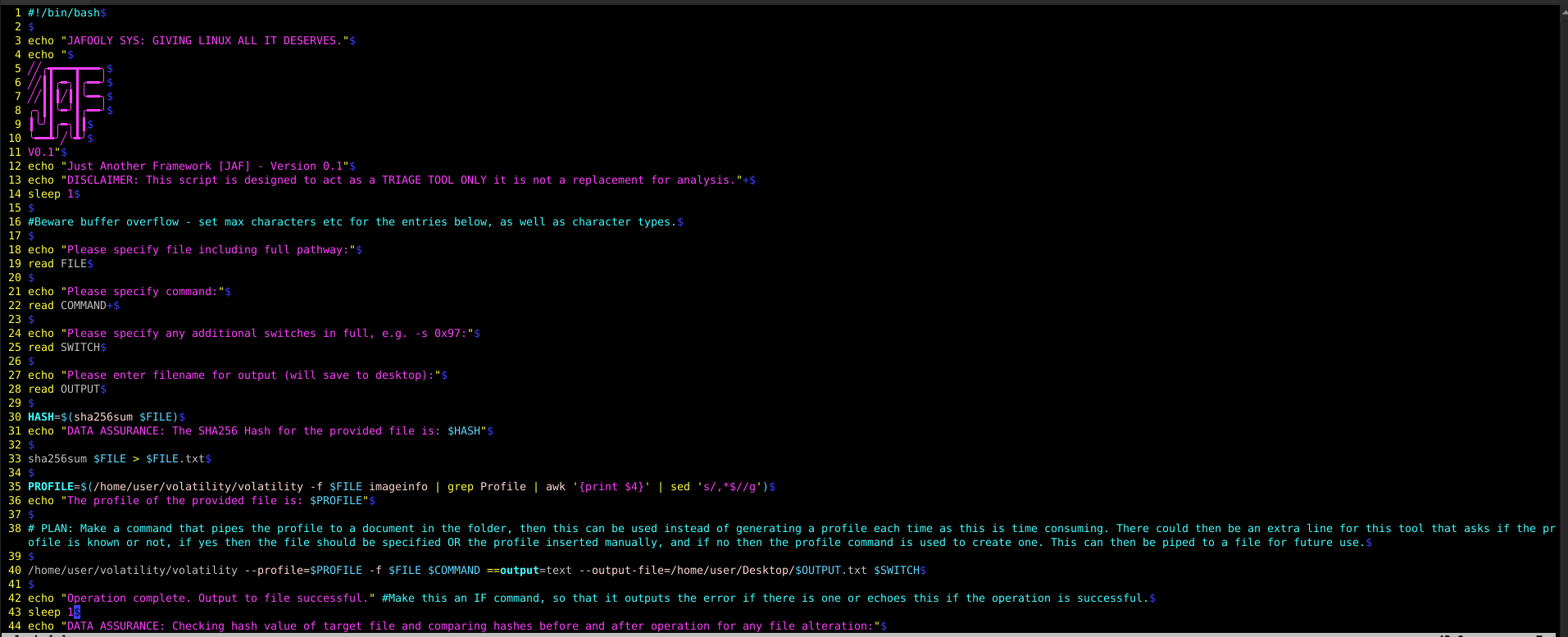
Additional functionality at this stage:   
- Security against buffer overflow by adding controls to entry on command line

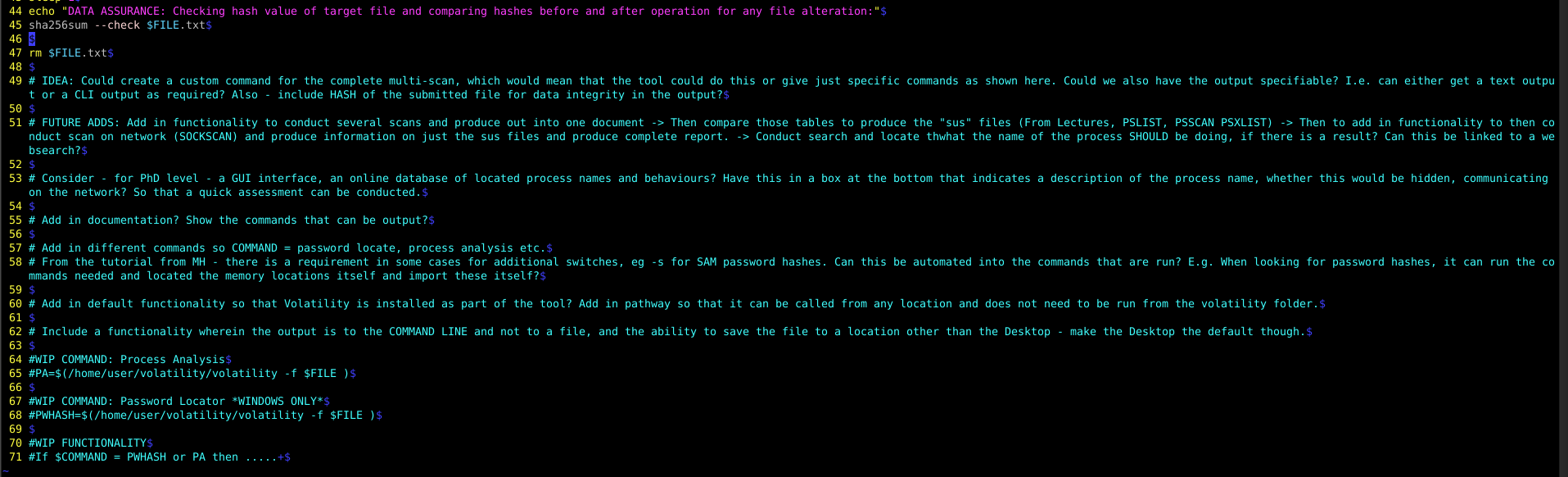
- Add in a profile command to allow user to add in profile if known, or create file with profile in if not run before so that the profile command does not need to be run each time. This will save time.

- Add in # comments to the commands to show what is doing what.

- Further functionalities added as comments to the script for continuous working.

- Meeting schedule with MH agreed and arranged.

SCRIPT STATE:   




**250521 – V 0.1**

Proposal amended to structure and reduced to word count maximum of 1430 – 10% over 1300.

Sent to MH for review and approval.

Created data assurance checks via the sha256 commands and tested.

**260521 – V 0.2**

Questionnaire created for DM and Cybercrime.

Researched and developed user input validation for the prompts within the script to protect against vulnerabilities such as buffer overflows and other issues such as syntax errors.

Meeting (online) with MH – Covered the existing functionality of the tool and the ideas for development.  
Advised to try the DIALOG option to create a better interface and cleaner output for the user that will aid the user.   
Also discussed how to create the bespoke commands that are required and what the functions will be.

Command created for partial password hash extraction:

*./volatility hivelist –-profile=XX -f XX | grep SAM | awk ‘{print $1;}’*

Created bespoke password hash extractor, and tested.

Began exploring the use of the *dialog* tool to create a UI for ease of interface.

CHANGES SAVED AS VOL Version 0.2

**270521 – V 0.3**

DECISION: The author does not like the way that Dialog presents the boxes that it creates. An alternative presentation solution will be found.

Added in character validation for user inputs. Selected accepted characters: ALPHANUMERIC, “.”, /, and \_ as these are present in most locations within Linux. Command changed to read rather than echo and user input.

Added in hidden process bespoke command and tested.

Created alternative “Test” script with menu commands contained and altered.   
DECISION: Propose both to DM and team and obtain preference.

CHANGES SAVED AS VOL Version 0.3

Changed name of script to jaf.sh

**280521 – V 0.4**

Created PROFILE commands so that the user can specify whether they know the profile or not, if yes then they are prompted for the details or a file containing the profile details, if no then the tool generates the profile and offers the user the choice to save this or not.

Created menus and nested menus for the above.

Amended previous read commands to allow user to backspace and correct errors without having to restart the script.

MEETING WITH DM AND CHESHIRE CYBER CRIME TEAM:   
Demonstrated the use of the tool.

Feedback was good – they stated that they can see the application of the tool and its uses.

They will complete questionnaires and get these back to me.

Obtained real-world sample from a machine that was running malware live at the time of obtaining the image.   
ISSUE: The size of this sample is 33GB which is more representative of the size of samples that are obtained by the unit in the field. When analysing this sample, the time taken for HASH and PROFILE functions is significant – over 20 mins. Work will be done to attempt to amend this.

Added in time measurements for each stage of the operation for testing monitoring.

CHANGES SAVED AS JAF Version 0.4

**290521 – V 0.4.1 & 0.4.2**

BUG FIXING: HASH function was being run twice, as shown by the delay in proceeding the script to profile generation after HASH creation. Located issue in script and removed additional function so that HASH is only run once.

TIME TRIALS:  
Multiple tests run to ascertain time to HASH using SHA256 the Dell malware sample file at size of 33GB:







(This was conducted on a DESKTOP machine)

Further delays are being experienced with profile generation. This has been tested as the delay is caused by the VOLATILITY imageinfo process, rather than an issue with the script.

This was left for 30 mins in script and did not complete, and was left for 30 mins in imageinfo from CLI rather than jaf and did not complete.

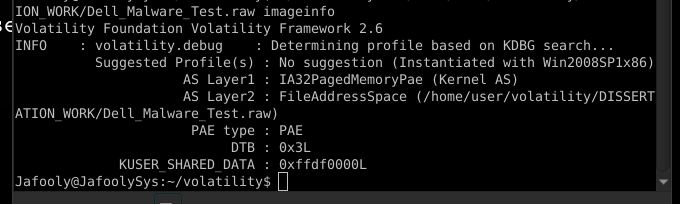
Script left running to locate completion time:



This also resulted in a profile of NO. Which is invalid.

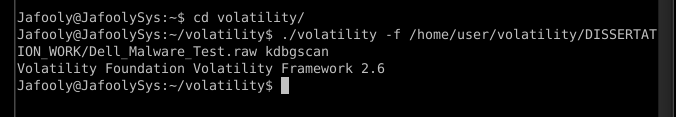
In order to troubleshoot this, the imageinfo command was run independently in order to ascertain the CLI output for review:

(18:57 begun)- The result of this scan was as follows:



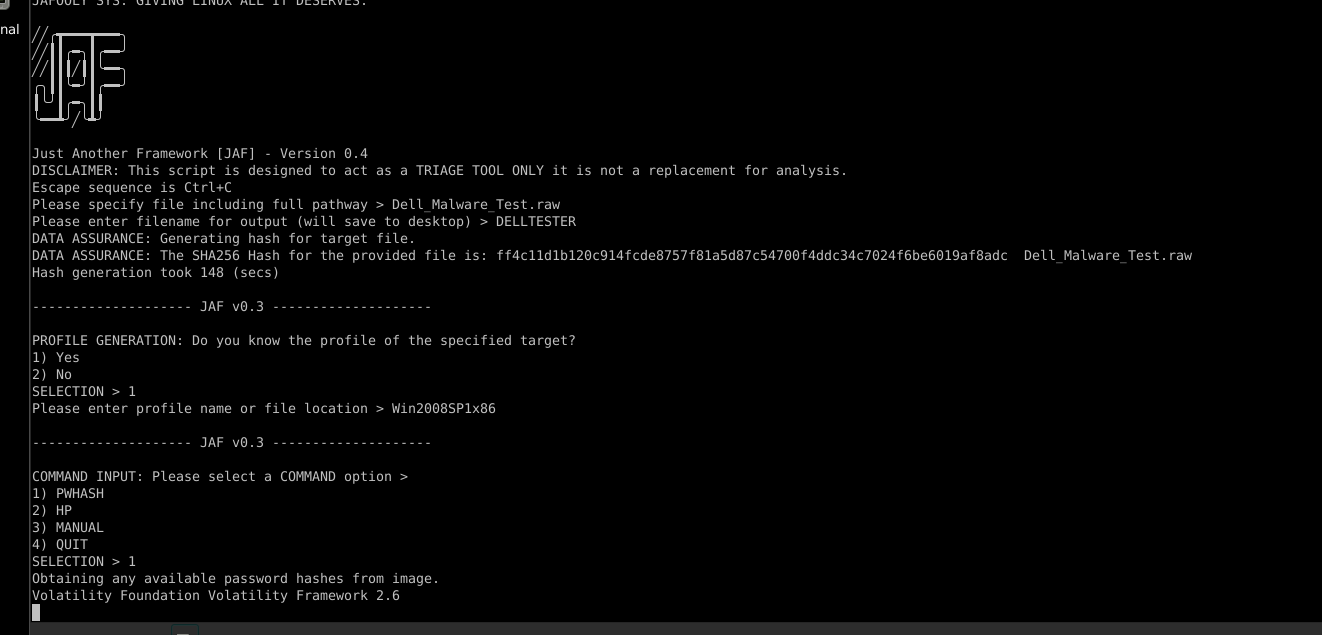
At the same time as running this, kdbgscan was also used to determine additional information that could be used in the generation of a profile:

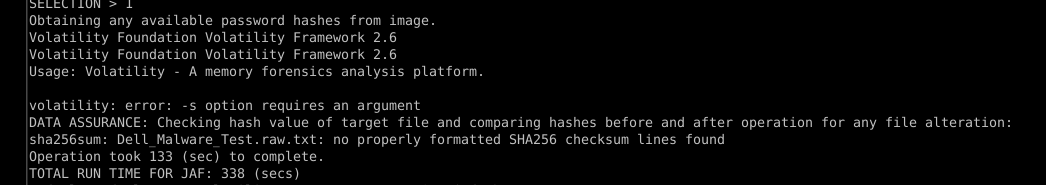
(18:58 begun) – The kbdg scan on this failed:



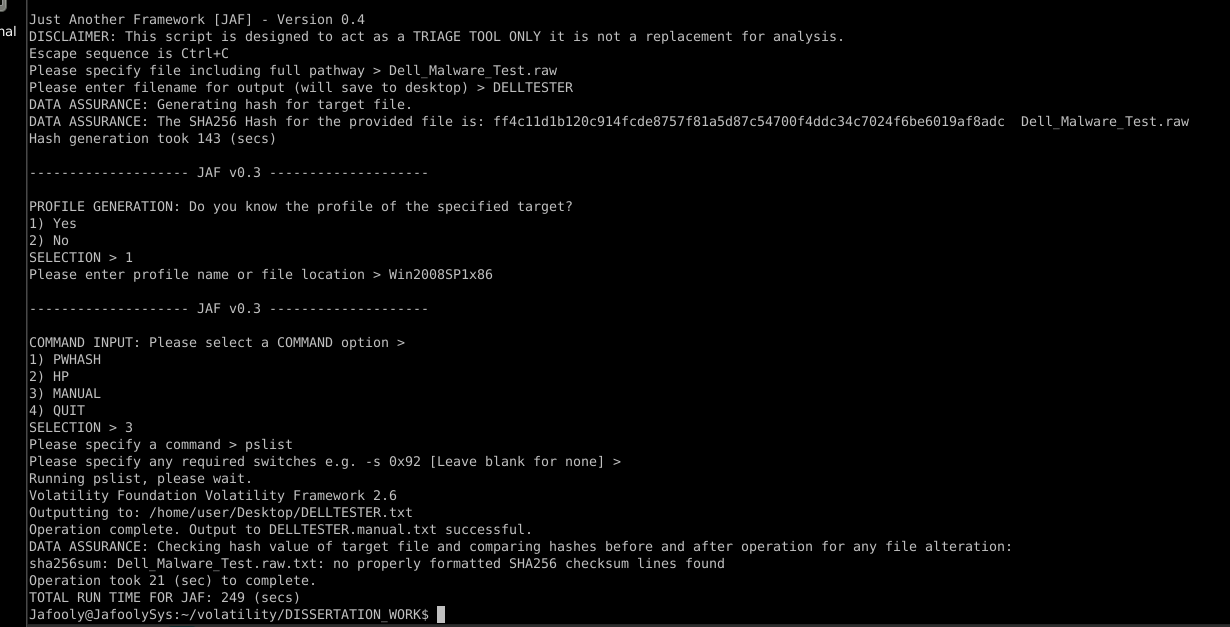
[This variation was done using IMAGEINFO to locate the profile, this could be problematic as multiple profiles could be located. KDBGSCAN should be used in this case – this will require a specific scan to be created to generate a profile!]

From a review of the imageinfo output, it appears that Volatility was struggling to locate a profile for the image.   
The reason for this was unclear.  
In order to test the profile that was used, jaf was run wit the provided profile of WIN2008SP1x86.   
The PWHASH command was used to test the output of this:





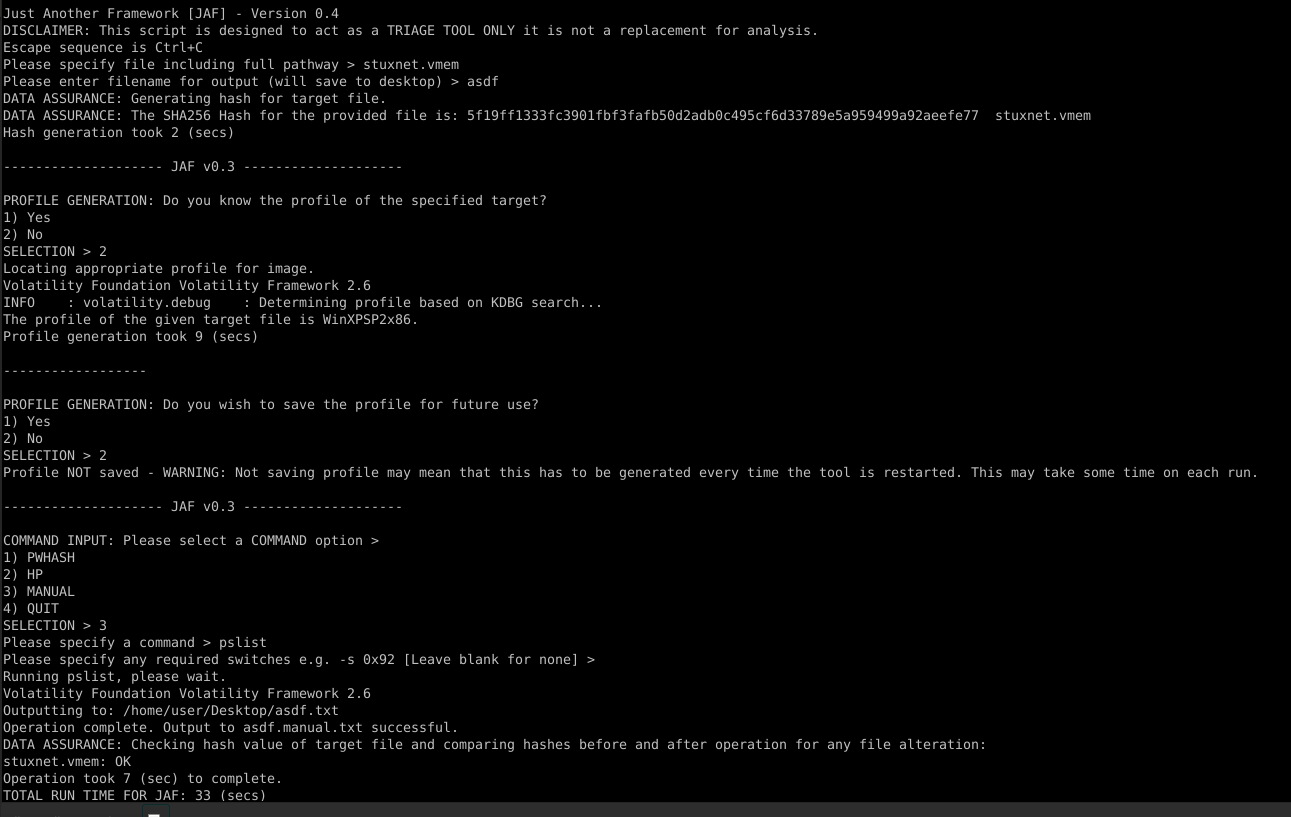
It is not clear what caused these errors, it could have been that the initial command failed and therefore the switch could not be completed as the memory address was never obtained.  
In order to test this further, the test was repeated with a simpler command to record what, if any, error or output was present:



Although this output completed, there was nothing in the file that was created.

It is suspected at this point that the errors may be with the file itself, and the extraction process, rather than the framework and its function.

To confirm this, the function of the tool in a previously used sample was checked, in this case: Stuxnet.vmem:



As can be seen, the tool is functioning correctly.

Therefore, the input file must be the issue. This is not the issue of the tool itself, this is an issue with Volatility as seen by the fact that use of the volatility python script itself also does not function.

MORE TESTING ACROSS A RANGE OF SAMPLES IS NEEDED.

IDEA FOR TIME ISSUES: Could ASSUMPTION be used along with the information that is known about the target system to cut down the time? I.e. specify IF functions to locate potential profiles and test these? What happens if the wrong one is used? Does volatility run or does it throw an error?

**310521 – V 0.4.3**

Added in IF command to profile generation to address the profile section failing – this will default to trying a KDBGScan to obtain the profile and will automatically carve out the profile from the result.

Added sus process command to link HP and sockscan.

Added malfind command

Changed files that are created to OUTPUT name rather than FILE

Saved changes as Jaf v0.4.3

Researched Bash GUIs

**010621 – V 0.4.4**

Implemented ZENITY GUI interface into script.

Added in directory creation and storage.

Added in optional saving of HASH information for integrity purposes.

Added in option to save or delete directory and files.

Saved menu version as JAF 0.4.4.

Tested functionality. Working.

Received first Questionnaire response from DS MACFARLANE. Awaiting further responses from other team members and officers.

**020621 – V 0.4.5**

Added README document and JAFINSTALLATION script. \*\* SAVE TO FILES  
(README – 0.4.5 refers) (JAFINSTALL – 1 refers)

Added in optional KDBGSCAN rather than mandatory.

Added in completed SUSPROCESS command with HP + SOCKSCAN

Changed version to V0.4.5 to reflect changes.

Sent ethics forms to MH for approval, along with information sheet and questionnaire copy.

**030621 – V 0.4.6**

Meeting with MH #1 – Supervisory meeting.

ACTIONS (Proposal):

Resources – replace Author with researcher.   
*ACTIONED - 030621*

Deliverables – Add contribution of this research, explain the idea and how that it benefits people rather than the automation itself.  
*ACTIONED - 030621*

Then proposal ready to submit. Awaiting Ethics forms from MH – he is reviewing.  
*SUBMITTED: 030621*

TOOL:   
Automate the process of profile generation – ask if the user wants us to guess, kdbgscan? Or do they want to see imageinfo themselves?  
Do the kdbgscan in the background and pick the most likely profile, then run some tests and present the user with the most likely.  
Only present this to the user when there is a key difference.

More commands.

* ACTION –

Added HPDLLDUMP command with automatic hidden process carving.

Saved as v 0.4.6.

**040621 – V 0.4.7**

Added clarity to the meaning of the directory creation following MH feedback.

Created PROFILE guessing command and saved as v0.4.7.

Amended README for new HPDLLDUMP command and profile guessing operation, as well as guidance for analysing MAC and LINUX files.

NOTE: The profile guessing command will require extensive testing. Multiple samples will be required across a range. **FOR DISCUSSION WITH MH AT NEXT MEETING RE CAN THESE SAMPLES BE PROVIDED BY THE UNIVERSITY?**

**080621+090621 – V 0.5**

Repaired bug in profile code that threw up syntax errors. \*\*

Added in function for volatility rather than explicit use and added commented out versions for both vol.py and volatility standalone uses to improve usability and range of compatible installations and systems.

Improved comments sections to make it clearer what each section does.

Saved version as 0.5 as PROFILE function now operates correctly.

**TO DO NOTES:**

Vol2 or 3?

Make sure that volatility is the latest version, get it to run the vol.py from wherever and alter the script so that it does this with the latest version etc.

Make it so that JAF is added to PATH so that it can be called anywhere, the same with volatility – change it so that volatility is autoinstalled and added to the pathway so that it can be called as VOL and then change the jaf script so that this is done.

Change manual save so that it is saved as COMMAND.MANUAL

Add in a “Have you done this before? OR does it need a new directory?”

Add in user spec for the directory location!

Need to see how to add Win10 profiles to Volatility !

Need to add in command to filter out useless profiles from kdbgscan

SOCKSCAN command

NETSCAN COMMAND

IE HISTORY COMMAND

REPORT OUTPUT FORMAT

PROFILE ASSUMPTION IDEA

Look at <https://www.hackingarticles.in/memory-forensics-using-volatility-framework/> and <https://resources.infosecinstitute.com/topic/memory-forensics-and-analysis-using-volatility/> put some of this information in!!

Find the lecture with Maryam where we spoke about the Stuxnet file and how Stuxnet was located and build this into a command !

Make it so that the command opens a new terminal window and keeps it open.

Need to come up with a list of commands etc that can be conducted in Volatility.

Need to come up with a way to use several commands at once?

Add in a progress bar?

Hide the volatility messages? Only my output?

How can we make it so that blank will output to the command line?

Add in functionality to conduct several scans and produce the output into one document -> then compare those tables to produce the “sus” files (From Lectures) -> Then to add in functionality to then conduct a scan of the network usage (SOCKSCAN) and produce information on just the susp files and produce a complete report. -> Conduct search and locate what the name of the process SHOULD be doing, if there is a result? Can this be linked to a web based search?

Online database of located process names and behaviours?? Have this in a box at the bottom that indicates a description of the process name, whether this would be hidden, communication on the network? So that a quick assessment can be conducted

Add in documentation to show the commands, what they do etc.

Add in a help function, like a man page. Can this be called immediately?

Add in default functionality so that it will install volatility by itself? Add in a pathway so that it can be called from any location and does not need to be run from the volatility folder.

Create installation documentation / script?

Add in that if the output is blank, then it outputs to the CLI?

Add in timestamps to the outputs

Silence the Volatility messages

Make it so that several commands can be specified (menu?) and run at once so that the user has minimal interaction

Add in a profile functionality so that the profile command does not need to be run every time as this is time consuming – this could either be given by the user, or created and stored in a file for future use ?

Read the Art of Mem Forensics book – look at the functions that are in there – IE HISTORY????

COMMAND MENUS:

1 menu for custom commands, and another entry that will lead to input for a general volatility command ?

TESTING:  
Initial questionnaire