
Windows Forensic Tool

Silver Bullet — Modernized

Project Report · Applied Computer Science

AUTHOR
Jesse Dumoulin

PERIOD
February 2026
– May 2026

Foreword

This report describes my internship project at i-Force where I worked as a forensic analyst. My project was a Windows forensic tool able to mount images, extract artefacts from them and generate reports based on the data gathered from these artefacts.

Reading this report will inform you about my journey — starting from analyzing the legacy tool to gain background information on my final objective, coming up with my own design, followed by the implementation of my design choices.

AI disclaimer

AI has been used to review and rewrite content in a more academic format. All content written was authored by Jesse Dumoulin because it is my own experience and project.

AI was also employed to generate the layout of the document.

Abstract

During my internship at i-Force I created a tool to assist investigators with the analysis of Windows images. The company uses a legacy tool called Silver Bullet, an in-house program created by an ex-employee. This tool became outdated and is hard to patch, so the company asked me to create an updated version from scratch.

The main objectives for this project were to mount the image, extract artefacts and generate a report from the gathered data. There were some additional artefacts i-Force wanted to be extracted from the image besides those in the legacy tool.

My tool was able to meet their objectives by employing a modular design to allow later development of new modules to extract new artefacts or update existing modules.

Table of Contents

Foreword	3
AI disclaimer	3
Abstract	4
Table of Contents.....	5
List of Abbreviations.....	6
Internship Environment.....	7
Company	7
My Role	7
Project Description.....	8
Objective.....	8
Legacy Tool Analysis	8
Design Philosophy	8
Comparison with Legacy Tool.....	9
Implementation.....	10
Outcome	11
Technical Difficulties	12
Development Timeline	12
Version 1 — Docker Container Approach.....	12
Version 2 — CLI Tool	12
General Difficulties	12
Timeline CSV Size.....	12
Module-Specific Difficulties	13
VSS Shadow Copies	13
Prefetch Module.....	13
Reflection	14
Conclusion	15

List of Abbreviations

Abbreviation	Full Term	Definition
DFIR	Digital Forensics and Incident Response	A specialized area of cybersecurity that focuses on identifying, investigating, and mitigating cyber incidents by collecting and analyzing digital evidence.
OSINT	Open Source Intelligence	The act of collecting and analyzing information about certain individuals or companies online through public sources.
TUI	Text User Interface	A program that runs in the terminal, using menus and keyboard navigation instead of typed commands.
VSS	Volume Shadow Copy Service	A Microsoft Windows feature that creates incremental backups that include old versions of files on the system.
CLI	Command Line Interface	A user interface in which you type commands instead of choosing them from a menu or selecting an icon.

Internship Environment

Company

I-Force is a company located in Erpe-Mere that specializes in fraud investigations, incident response and digital forensics. The company has two departments: the fraud department and the IT department. Both departments work together closely with each other for some cases, the investigation for fraud cases will be led by the fraud department but they rely on the IT department to further investigate digital traces. This is reflective of the current state of the world where many things are done digitally making the IT department a valuable addition to fraud investigations.

Both teams are quite small in size and the amount of employees add up in the ten to fifteen range. The IT team does most of its work in English because half of the team is international. The whole team is welcoming and the office provides a positive working space.

Every Monday, the team holds a meeting to discuss active cases and review their progress. Some cases are bigger than others and these meetings give other employees who are going into the week with less work the opportunity to help with the bigger cases.

My Role

My tasks during my internship are helping with cases (either fraud or IT) or work on the internship project. We use a wide variety of tools for many different purposes ranging from OSINT to developing new tools to use for forensic investigations. I also had the opportunity to learn by reading old case reports and by watching how my senior colleagues handled certain things. While working I can rely on my colleagues to explain things I didn't understand making the office not only an environment for work but also for learning which actually ties back to one of the core values of I-Force which is continuous learning.

My work was always performed on site since the company uses a specific network to access current case files and reports. My assistance on cases was requested by the person requiring some assistance while working on their case. When not working on cases I was working on the project or reading some old cases.

Project Description

Objective

The project I chose is called “silver bullet”. This refers to a legacy forensic tool mainly used for post-employment screenings which i-Force wanted to update. During the explanation of what the project is about they told me to make it from scratch and not to patch the old tool. Their reasoning for this was that browser history did not always work, the USB analysis was incomplete and problems with mounting the correct path on edge cases such as empty bitlocked or weirdly partitioned disks.

The tool focusses on extracting windows artefacts and generates a report and timeline which can be used in the case report and for further investigation by the forensic analyst. The tool is supposed to make the job for the forensic analyst easier by automating extraction of artefacts and parsing their information in a format that makes analyzing easier.

The objective of the project is thus to develop a tool that mounts forensic images of windows systems, extracts artefacts from the mounted image and then outputs the useful information to a report and timeline for later analysis.

There are also some new features i-Force wanted me to implement while making the new tool. The requested features include recycle bin forensics, querying the windows installation date and also exporting deleted files from the current system but also permanently deleted files that are recoverable from vss.

Legacy Tool Analysis

Before beginning with development of my project, I started by analyzing the old tool. The legacy tool is made in the python language, a main script serves as the entry point for the tool.

On startup, the tool first asks for a case number and evidence number, then it asks for the forensic image. After the image is mounted it starts calling subscripts and outputs artefacts to a report in html and some artefacts to a csv file to create a timeline. The whole tool runs as a monolithic script running completely each time it is ran. I found some points for improvement here. First of all some artefacts are only required for further investigation making the tool take more time especially with bigger cases. Secondly if the tool, or system it is running on, would crash it would need to run again from the start to make its report. These observations were kept in mind for when I would start developing.

Another important part of my analysis was reading old cases. By reading old cases I was able to learn more about how different artefacts are used in investigations, what they prove and what information I should also extract.

Design Philosophy

For the design of my tool I decided to use bash scripts to simplify calling upon different extraction tools for artefacts. The decision to run the tool as a bash script also makes mounting forensic images and unlocking BitLocker partitions easier.

The structure of the tool itself can be split up into module scripts and infrastructure scripts. Infrastructure scripts can make use of module scripts to perform tasks such as extracting or parsing artefacts but they are also responsible for setting up the environment so the module scripts can run.

Module scripts come in pairs, one is responsible for extracting another one is responsible for parsing the artefacts and adding the relevant information to the report and timeline. This makes adding or modifying a module only take two files unless the specific modules needs a new argument then we make some other changes in an infrastructure script. However, if an artefact would add new information or change its structure we only need to modify two files.

My decision to separate tasks between infrastructure and module scripts improves maintainability over time by making debugging easier should a problem come up at a later point in time. By making modules separate entities we also prevent that a failure in one module would break another.

Comparison with Legacy Tool

Comparing my project to the legacy tool, there are a couple of differences. The biggest difference is the modular design of my project, because of this decision I create better maintainability and flexibility. To make changes to the legacy tool I would need to create scripts and make changes to the main script. Making changes to this main script becomes chaotic over time as more features are added to the tool whereas using my design creating a new feature is isolated from other modules.

Another difference is that the legacy tool could only be used for data exfiltration cases, there might however be some cases where using the legacy tool can be useful but certain data would be missing and the analyst still needs to extract it themselves. My tool can be modified with ease giving it a wider variety of use cases depending on the selected configuration and the ability to add or create custom configurations.

For output both tools generate a readable version. The html provided by the legacy tool and markdown by my tool. This provides a general idea of the system's artefacts and is useful for management summaries. Also using csv for creating a timeline is easy to work with and can be filtered using tools such as excel.

Implementation

The tool is user friendly because of the TUI layer for most actions. When starting the tool it first asks for case number, evidence number and uses a TUI to select the path to the forensic image. During the mounting phase it checks if the system is encrypted using BitLocker and asks for decryption method to make the system accessible. When the system is available for us to use the tool finds the current control set of the system and asks the user to select the user to be investigated then the forensic shell is started. The forensic shell is also a TUI displaying three options to the user: exit, config and run.

Selecting exit unmounts all evidence files from the system and exits the tool. The cleanup is implemented here so the system we use does not have mounted partitions from old cases due to human errors.

Selecting config presents the user with multiple preset configurations and a choice to make a custom one themselves. A configuration is just a list of modules that will run before generating a report. For my project I decided to create a fully functional data exfiltration configuration that uses many different artefacts to look for signs of data exfiltration. Further development can be done to create different configurations for different use cases.

Selecting run starts the gathering of selected artefacts using the list of modules selected by using config. When the tool has finished gathering the artefacts requested, report generation will commence. To generate the report the tool finds the names of modules in an output directory and calls the script to parse the data from the module it found. I use this method to ensure all artefacts that were gathered from the evidence are being reported on in the report, even artefacts gathered in a previous runtime.

The generated report includes a text based report in markdown and a csv timeline that can be imported into excel. Certain modules have their own form of output such as recycle bin forensics outputs the recovered files and event logs generate separate csv documents because of the large amount of data extracted.

Another feature I implemented in the tool is internal logging to keep track of events. Examples of events are starting the tool, starting a module or starting report generation. We use this to prove when an analyst used the tool and for what purpose.

When developing modules, I first started with doing research on what the artefact will prove, what tools are available for me to extract them. Next I start with making the gathering script, from there I go through the generated output to figure out how I would present relevant data in the report or timeline. Finally I start working on the reporting script and through trial and error I get to the final version of the module.

Outcome

The final version of my project is able to assist investigators in a timely manner with their investigation by mounting windows images, gathering artefacts from them and generating a human readable report from the gathered data. The purpose of the tool is not to draw conclusions but to speed up the time it takes to finish a case.

My project can provide the same data as the legacy tool and more. The usb artefacts are a good example of this because the legacy tool pulled artefacts from one registry hive whereas my tool pulls artefacts from four different sources to have a more detailed report for the investigation. This means that when comparing the legacy to the new tool, the new tool would have the same data, more detailed information and additional artefacts ready to be analyzed, leading to a better informed final report by the investigator.

My tool is able to recover files from the recycle bin from both the live system as well as all vss shadow copies as requested by i-Force when the project had started. I also added some additional information about the operating system such as the version of windows running on the system and when the latest install of windows happened. With this all of the objectives i-Force had set have been fulfilled.

Further development of the project would focus on looking into fetching information from windows system files, both from the image and all of the vss shadow copies, to learn more about them and potentially find deleted evidence.

Technical Difficulties

Development Timeline

While working on my project I experienced some difficulties which led to different versions of my tool. My first challenge was the restricted permissions on the laptop provided for my internship making mounting a forensic image already a hurdle to overcome. To solve this problem I started working on my first version of my tool.

Version 1 — Docker Container Approach

The first version I made used docker containers. This was my way to circumvent the problems with mounting images on my laptop. This version of the tool included the complete mounting script which is still used in the final version of the tool. The foundation for the modular design was already established, however some problems came along with this version such as installing tools in the docker container was unreliable as the compilation failed sometimes. The tool was originally meant to be used in a SIFT workstation (an ubuntu based forensic linux distribution) but because this wasn't deployed yet I used this version as the groundwork when migrating to my next version.

Version 2 — CLI Tool

The second version was a mostly CLI tool requiring the user to type out the full path to the image when mounting and using commands such as "run eventFiles" to run a module. The version itself is still in use but I added the TUI layer on top of it to reduce human errors and increase efficiency.

General Difficulties

Timeline CSV Size

During development I noticed that my timeline csv became too big for excel to handle without issues. To remediate this problem I made a power BI template that can load the csv and give visual representations of the data found in the timeline. This template is reusable because the structure of the csv will always be the same for each case making it a well fit solution to the problem.

Module-Specific Difficulties

VSS Shadow Copies

After I finished development on the vss module, I ran a test on an old case that contained three shadow copies. My tool failed to find the shadow copies and crashed the complete tool in the process. The tool crashed due to an error being thrown as a failsafe so fixing this issue also fixes the tool crashing.

My first action was to investigate the image using Arsenal Image Mounter to see if the shadow copies actually exist. After confirming this I looked at the documentation of the tool I use to identify the shadow copies to find if there have been any changes that could cause this error but nothing should cause this error.

I started thinking about the differences between both cases. The first has two copies, the second has three but this does not impact the discovery of the copies. There was one thing I hadn't thought about and that is the way both cases are mounted. The first case is not bitlocked meaning that there is no encryption on the disk, the second one is Bitlocked. The raw image is used as source to discover the copies but because the encrypted image is well encrypted the copies can't be found.

I introduced a new variable called `vss_source` that changes depending on whether the raw image was encrypted or not. This fixed the issue, now my tool can identify shadow copies on both Bitlocked and non-Bitlocked images.

Prefetch Module

I identified an issue when generating the report for prefetch on an old case. There was an invalid prefetch file and errors were redirected into the output file directly instead of first being handled. This broke the parsing of the output file. I changed the error handling for the gathering part of this module so the output structure doesn't break anymore.

Reflection

When starting my internship at i-Force they proposed six different projects for me to choose from. I chose the silver bullet project because it had a lot of potential to be something I could learn a lot from during development. It also was the only project directly related to forensics which is a field where there is much to learn and also my biggest interest.

During my internship I also assisted with some cases. Because of this development sometimes stagnated for a while working on the case or writing reports. Sometimes this break from developing allowed me to come back to the project with a fresh mind and help me find solutions I didn't see before.

Having experience with linux machines allowed me to work efficiently on the project by having a better understanding of the different commands I can use to complete certain tasks.

While working on the project I was able to learn a lot of new things about windows specifically such as What kind of artefacts can be found, What are they used for, What do they prove. This opened up my interest into malware analysis more because now I want to figure out What is the malware abusing, How is it abusing it, How can we stop it.

Something I would do differently is performing more research into different tools I can use for different artefacts, during this project I usually worked with the first tool I found that could do the job until it had to be replaced by something more detailed.

Conclusion

My internship and project provided a hands-on experience in DFIR and I was able to learn a lot about windows systems in particular. Learning about how a report is made was one of the toughest things during my internship because you need to understand that what you write down will have consequences.

While developing my project I first analyzed the legacy tool. By doing this I noticed that a modular structure would be a better fit for my project because it allows i-Force to add or modify artefacts to be added into the investigation without having to make changes to the main infrastructure of my tool. The tool is able to fully mount 500+GB images and extract artefacts from them before generating a human readable report from the extracted data. This report is then later used by investigators working on the case. During my development I changed the tool to a TUI for a better end-user experience.

My project is able to cut down the time it takes an investigator to analyze an image noticeably when comparing to the legacy tool and doing a manual extraction of artefacts.

