Final Report

Ariel Mitchell

Cameron Walters

12/3/25

IT 360 Project Report

Introduction

We created a digital forensic tool that allows an expert to investigate user-focused artifacts, focusing on login history. The purpose of this tool is to provide experts with information including time stamps of logins, duration of login sessions, and active sessions. Automating a task that is done repetitively saves time and reduces human error. We felt it was important to create a tool that aids in the investigation process while ensuring it will be forensically sound. Ultimately, this project aimed to provide investigators with an accurate, effective and field ready tool that will assist with their investigation process towards systems login logs.

Technical Implementation

Our project was implemented using a Bash shell script that was designed to gather and log all user authentication events on Linux systems. The script leverages native system utilities and logs to ensure integrity and reproducibility of the data collection process. Bash was chosen due to its prevalence across Linux systems and for direct interaction with system logs and utilities.

We used the commands last, journalctl, grep, cat, and sudo. Last allows for successful login events to be extracted from the /var/log/tmp file. This reads from the log file without modifying data and preserves all original timestamps and other relevant information. Journalctl is used to query the systemd journal logs for three criteria of authentication related events. Grep was used

to filter the journalctl query for specific keywords such as “invalid user.” This allows for extraction of relevant events while keeping other events away from the output file. Sudo was used to elevate access for the logs that were protected. Finally, cat was used to display the final output to the user.

We used a variety of techniques while building our script to ensure that it remained forensically sound. All collected output was written to a single output file using “>” which is a subshell block. This prevents partial writes which in turn preserves forensic integrity. “--no-pager” is used with our journalctl queries to disable interactive pagination which was used to ensure that full log output is captured without truncation. “2>/dev/null” was used to prevent irrelevant stderr messages during the grep commands to maintain clean output without altering log content.

All commands in our program operate in read-only mode to prevent modification to original logs or the system state itself. No external dependencies are needed as the script relies solely on built-in Linux commands and utilities, avoiding any contamination from third-party tools. The final output file is displayed to the user and saved in plaintext, allowing for chain-of-custody documentation and peer review. Finally, the script does not install, modify, or delete any system components to preserve the forensic environment.

Results

Upon execution of the script a categorized report of authentication activity is created.The results are organized into three distinct categories that highlight different aspects of user access: Successful Logins, Failed Logins, and Local Authentication Attempts.

Successful logins will identify accounts that were successfully authenticated paired with a timestamp that records the exact date and time of the event. This is followed by the remote host or IP from which the login was attempted from. Terminal information is displayed to indicate whether or not the attempt was local or remote. The final piece of information tied to each event is session duration providing logout times or session lengths allowing for user activity timelines to be constructed.

The failed login results provide insight into all unsuccessful authentication events on the system. Each entry will include the username that was used, whether valid or invalid, paired with the exact timestamp of the event. The source IP address and associated port are recorded to distinguish between local and remote based attempts, such as SSH. The record will then include the specific reason for failure: incorrect passwords, non-existent accounts, or general errors such as too many attempts.

The local authentication results show all activity that occurred directly on the machine rather than remote connections. This section includes all records of sessions being opened and closed at the console, along with the usernames and timestamps involved with the event. Failed local login attempts are also recorded showing when users either entered incorrect credentials or tried to access with an invalid account.

Lessons Learned & Conclusion

Developing this script has given us hands-on experience in writing Bash scripts and a stronger understanding of how Linux systems record authorization events across different log sources. We were able to learn how successful, failed, and local authentication attempts are stored in separate places and how within these logs filtering can be used to reduce redundancy.

During the development of our program we encountered challenges with redundant returns of entries across sudo, gdm, and user authentication events. This required us to implement additional filtering to ensure no redundancy was present. Originally, our program contained an active sessions section, but it was found that there was an overlap as this information was concurrently pulled by the successful logins section. This section was ultimately removed to prevent duplicate events. Another change was the decision to remove Python from the program, as we determined that relying on only native shell commands was more forensically sound.

Overall, the script works well for organizing authentication information into clear sections for forensic experts, but working through development has highlighted certain areas that could be improved. In future iterations, we would try to focus on building more filters to better separate essential events from excessive information, as well as controls to allow customization of these filters on a UI for ease of use. This would make the tool more adaptable and reduce manual editing, allowing for investigators to quickly tailor output for their specific needs.