Table of Contents

Task 1: Log Analysis for Unauthorised Access

1. Methodology for log analysis

The log analysis process consisted of importing the provided access log file into CyberChef, which is a powerful tool for filtering and analysing text. The primary goal was to read the file and identify any signs of unauthorised activity, such as repeated failed logins, suspicious IP addresses, and irregular access patterns.

To efficiently filter the logs, a regular expression with the regex “Error” was applied, giving the ability to isolate the log entries to display all the failed access attempts. This method of filtering allowed the ability to identify HTTP 403 errors, which indicates a unauthorised access response from the server (Mozilla Foundation, 2025). Furthermore, the results revealed repeated unauthorised attempts targeting the “*/admin*” page from the IP address “*192.168.1.15”* over multiple days.

The logs were examined further, and there was a clear and consistent attack pattern of the IP address making unauthorised access attempts at a precise three-day interval at the same time. Such a structured pattern strongly suggests that the attack was automated, likely with a scheduled script, rather than the attempts being multiple manual access attempts. This strongly indicates that the attacker used a brute-force approach to finding vulnerabilities within the system.

Correlation with Task 2: USB File Access
Further investigation revealed that the same IP address *(192.168.1.15)* was also within the USB device logs file, which could suggest a connection between the unauthorised access attempts. There is an interesting pattern that was identified within the file log where the admin was logged into from the IP addresses increasing by five each time, for example, *192.168.1.10, 192.168.1.15, 192.168.1.20* and so on, which started from *192.168.1.10 to 192.168.1.50*; this systematic increase could suggest that there is another automated script or a dynamic IP allocation technique. This could indicate an attempt of anti-forensic techniques to evade detection or bypass IP restriction.

2. Key findings

TIMESTAMP	USER ACCOUNT	IP ADDRESS
2024-02-01, 12:34:56	admin	192.168.1.15
2024-02-04, 12:34:56	admin	192.168.1.15
2024-02-07, 12:34:56	admin	192.168.1.15
2024-02-10, 12:34:56	admin	192.168.1.15
2024-02-13, 12:34:56	admin	192.168.1.15
2024-02-16, 12:34:56	admin	192.168.1.15
2024-02-19, 12:34:56	admin	192.168.1.15
2024-02-22, 12:34:56	admin	192.168.1.15
2024-02-25, 12:34:56	admin	192.168.1.15
2024-02-28, 12:34:56	admin	192.168.1.15

3. Screenshots

Arranging the access error events within a timeline allows the investigators to identify potential unseen patterns and anomalies and notice the difference between an unortharised access attempt and file interactions. Utilising tools such as CyberChef regex filtering enables the ability to identify errors efficiently and document evidence within forensic investigations.

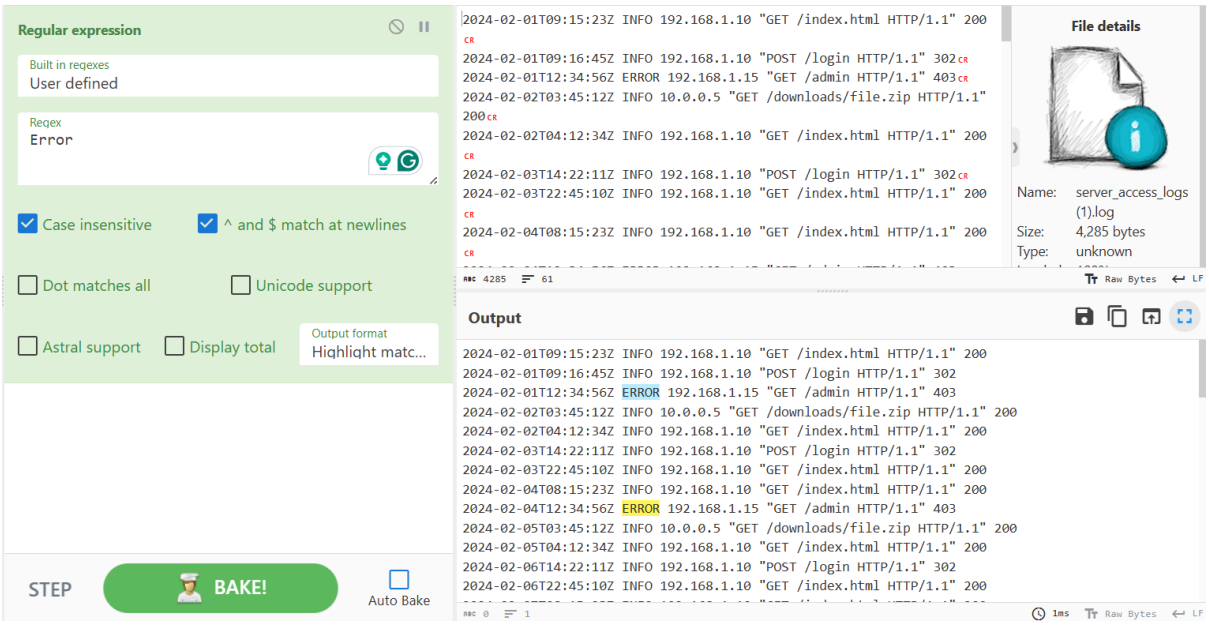


Image 1 – Analysis Tool used with CyberChef

```
2024-02-01T09:15:23Z INFO 192.168.1.10 "GET /index.html HTTP/1.1" 200
2024-02-01T09:16:45Z INFO 192.168.1.10 "POST /login HTTP/1.1" 302
2024-02-01T12:34:56Z ERROR 192.168.1.15 "GET /admin HTTP/1.1" 403
2024-02-02T03:45:12Z INFO 10.0.0.5 "GET /downloads/file.zip HTTP/1.1" 200
2024-02-02T04:12:34Z INFO 192.168.1.10 "GET /index.html HTTP/1.1" 200
2024-02-03T14:22:11Z INFO 192.168.1.10 "POST /login HTTP/1.1" 302
2024-02-03T22:45:10Z INFO 192.168.1.10 "GET /index.html HTTP/1.1" 200
2024-02-04T08:15:23Z INFO 192.168.1.10 "GET /index.html HTTP/1.1" 200
2024-02-04T12:34:56Z ERROR 192.168.1.15 "GET /admin HTTP/1.1" 403
2024-02-05T03:45:12Z INFO 10.0.0.5 "GET /downloads/file.zip HTTP/1.1" 200
2024-02-05T04:12:34Z INFO 192.168.1.10 "GET /index.html HTTP/1.1" 200
2024-02-06T14:22:11Z INFO 192.168.1.10 "POST /login HTTP/1.1" 302
2024-02-06T22:45:10Z INFO 192.168.1.10 "GET /index.html HTTP/1.1" 200
2024-02-07T08:15:23Z INFO 192.168.1.10 "GET /index.html HTTP/1.1" 200
2024-02-07T12:34:56Z ERROR 192.168.1.15 "GET /admin HTTP/1.1" 403
2024-02-08T03:45:12Z INFO 10.0.0.5 "GET /downloads/file.zip HTTP/1.1" 200
2024-02-08T04:12:34Z INFO 192.168.1.10 "GET /index.html HTTP/1.1" 200
2024-02-09T14:22:11Z INFO 192.168.1.10 "POST /login HTTP/1.1" 302
2024-02-09T22:45:10Z INFO 192.168.1.10 "GET /index.html HTTP/1.1" 200
2024-02-10T08:15:23Z INFO 192.168.1.10 "GET /index.html HTTP/1.1" 200
2024-02-10T12:34:56Z ERROR 192.168.1.15 "GET /admin HTTP/1.1" 403
2024-02-11T03:45:12Z INFO 10.0.0.5 "GET /downloads/file.zip HTTP/1.1" 200
2024-02-11T04:12:34Z INFO 192.168.1.10 "GET /index.html HTTP/1.1" 200
2024-02-12T14:22:11Z INFO 192.168.1.10 "POST /login HTTP/1.1" 302
2024-02-12T22:45:10Z INFO 192.168.1.10 "GET /index.html HTTP/1.1" 200
2024-02-13T08:15:23Z INFO 192.168.1.10 "GET /index.html HTTP/1.1" 200
2024-02-13T12:34:56Z ERROR 192.168.1.15 "GET /admin HTTP/1.1" 403
2024-02-14T03:45:12Z INFO 10.0.0.5 "GET /downloads/file.zip HTTP/1.1" 200
2024-02-14T04:12:34Z INFO 192.168.1.10 "GET /index.html HTTP/1.1" 200
2024-02-15T14:22:11Z INFO 192.168.1.10 "POST /login HTTP/1.1" 302
2024-02-15T22:45:10Z INFO 192.168.1.10 "GET /index.html HTTP/1.1" 200
2024-02-16T08:15:23Z INFO 192.168.1.10 "GET /index.html HTTP/1.1" 200
2024-02-15T14:22:11Z INFO 192.168.1.10 "POST /login HTTP/1.1" 302
2024-02-15T22:45:10Z INFO 192.168.1.10 "GET /index.html HTTP/1.1" 200
2024-02-16T08:15:23Z INFO 192.168.1.10 "GET /index.html HTTP/1.1" 200
2024-02-16T12:34:56Z ERROR 192.168.1.15 "GET /admin HTTP/1.1" 403
2024-02-17T03:45:12Z INFO 10.0.0.5 "GET /downloads/file.zip HTTP/1.1" 200
2024-02-17T04:12:34Z INFO 192.168.1.10 "GET /index.html HTTP/1.1" 200
2024-02-18T14:22:11Z INFO 192.168.1.10 "POST /login HTTP/1.1" 302
2024-02-18T22:45:10Z INFO 192.168.1.10 "GET /index.html HTTP/1.1" 200
2024-02-19T08:15:23Z INFO 192.168.1.10 "GET /index.html HTTP/1.1" 200
2024-02-19T12:34:56Z ERROR 192.168.1.15 "GET /admin HTTP/1.1" 403
2024-02-20T03:45:12Z INFO 10.0.0.5 "GET /downloads/file.zip HTTP/1.1" 200
2024-02-20T04:12:34Z INFO 192.168.1.10 "GET /index.html HTTP/1.1" 200
2024-02-21T14:22:11Z INFO 192.168.1.10 "POST /login HTTP/1.1" 302
2024-02-21T22:45:10Z INFO 192.168.1.10 "GET /index.html HTTP/1.1" 200
2024-02-22T08:15:23Z INFO 192.168.1.10 "GET /index.html HTTP/1.1" 200
2024-02-22T12:34:56Z ERROR 192.168.1.15 "GET /admin HTTP/1.1" 403
2024-02-23T03:45:12Z INFO 10.0.0.5 "GET /downloads/file.zip HTTP/1.1" 200
2024-02-23T04:12:34Z INFO 192.168.1.10 "GET /index.html HTTP/1.1" 200
2024-02-24T14:22:11Z INFO 192.168.1.10 "POST /login HTTP/1.1" 302
2024-02-24T22:45:10Z INFO 192.168.1.10 "GET /index.html HTTP/1.1" 200
2024-02-25T08:15:23Z INFO 192.168.1.10 "GET /index.html HTTP/1.1" 200
2024-02-25T12:34:56Z ERROR 192.168.1.15 "GET /admin HTTP/1.1" 403
2024-02-26T03:45:12Z INFO 10.0.0.5 "GET /downloads/file.zip HTTP/1.1" 200
2024-02-26T04:12:34Z INFO 192.168.1.10 "GET /index.html HTTP/1.1" 200
2024-02-27T14:22:11Z INFO 192.168.1.10 "POST /login HTTP/1.1" 302
2024-02-27T22:45:10Z INFO 192.168.1.10 "GET /index.html HTTP/1.1" 200
2024-02-28T08:15:23Z INFO 192.168.1.10 "GET /index.html HTTP/1.1" 200
2024-02-28T12:34:56Z ERROR 192.168.1.15 "GET /admin HTTP/1.1" 403
2024-02-29T03:45:12Z INFO 10.0.0.5 "GET /downloads/file.zip HTTP/1.1" 200
2024-02-29T04:12:34Z INFO 192.168.1.10 "GET /index.html HTTP/1.1" 200
2024-03-01T14:22:11Z INFO 192.168.1.10 "POST /login HTTP/1.1" 302
2024-03-01T22:45:10Z INFO 192.168.1.10 "GET /index.html HTTP/1.1" 200
```

Image 2 – Highlighted text shows login attempts.

Task 2: Evidence Integrity and Recovery

1. Methods, tools & recovered evidence

In this forensic investigation of the Microsoft Excel file (client.xlsx), multiple systematic and rigorous attempts were made to decrypt this file through wordlists and dictionary-based attacks without success. The main attempts included using two widely known word lists, such as *rockyou.txt* and *10-million-password-list-top-100000*. However, further testing also included tools such as Hashcat's rule-based permutations (rule 64), which was also unsuccessful, even when combined with a custom-generated wordlist using information from the *log* file for potential passwords.

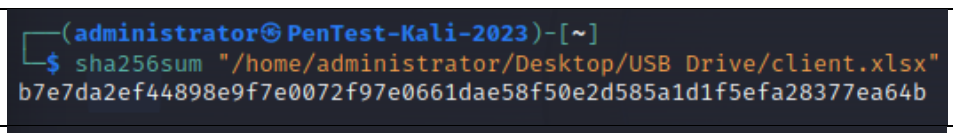
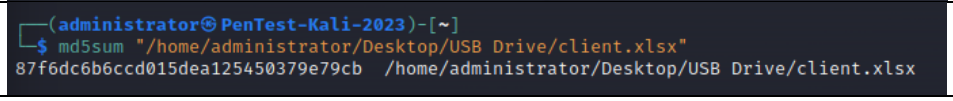
Due to the consistent failure of all of these attempts, it strongly indicates that the password is not included in a dictionary wordlist and is much more complex and highly secure. Microsoft Excel file encryption uses AES 256-bit encryption (NTNU, 2025); in turn, this means that encrypted files with large complex passwords, mixed character types, or random patterns are very unlikely to be vulnerable to dictionary word list attacks, even those with rule-based extensions.

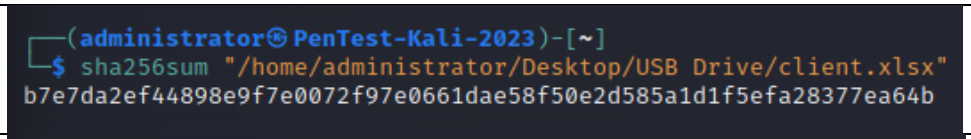
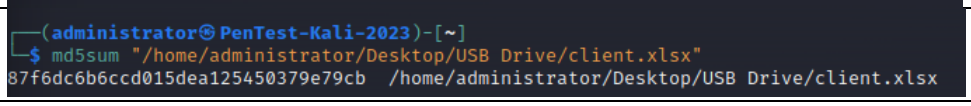
To further confirm that the file decryption was indeed resistant to password wordlists and dictionaries, Hashcat, as a brute force approach, was used to utilise the maximum allocation of GPU usage for efficiency. However, even when attempting to guess character cases, types, and sizes, the ETA was approximately 177 days to decrypt, assuming those variables were correct. This ETA with the current resources and deadlines is unrealistic and impractical for a forensic investigation.

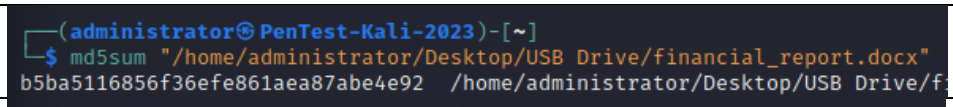
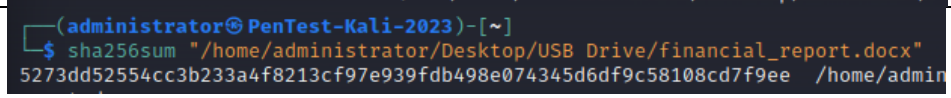
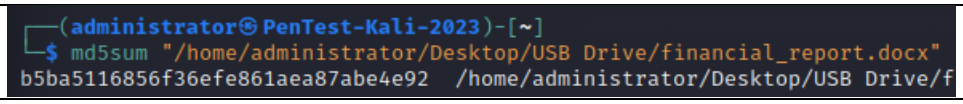
Using forensic best practices, the decryption of strong encryption methods can make it very secure and difficult for investigators to decrypt (XPressGuards, 2025). Therefore, allocating six months to brute force a single encrypted file is not realistic and justifiable, especially when time and resources are limited.

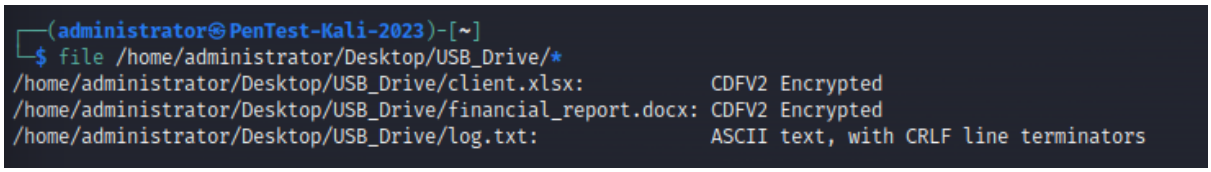
Overall, due to the impracticality of brute force methods, this Microsoft Excel file (*client.xlsx*) has to be considered not feasible to decrypt within the justification of time, thereby making it untrackable for forensic purposes within this investigation.

2. Summary table

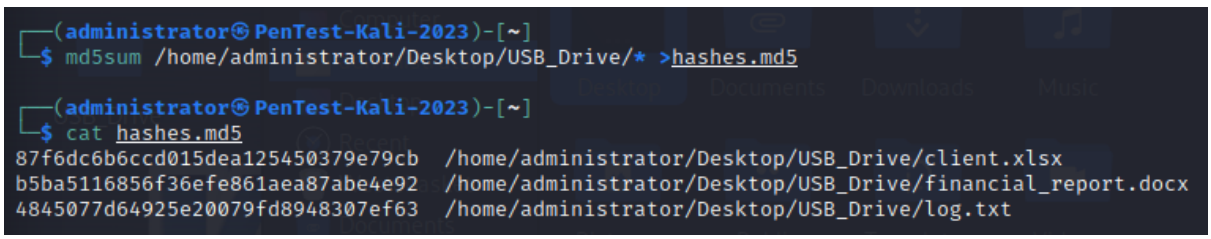
File Name	client.xlsx
Original SHA-256 Hash	
Original MD5 Hash	

Recalculated SHA-256 Hash	
Recalculated MD5 Hash	
Integrity Notes	The integrity of the file is verified as the hashes match; however, due to AES 256-bit and the time limit, brute force was not practical.

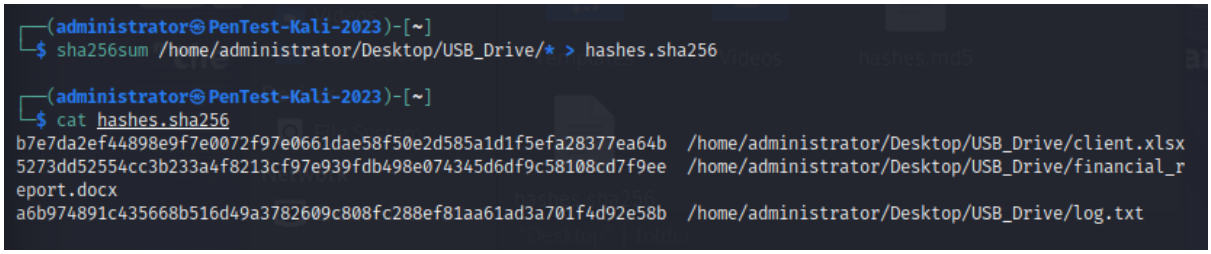
File Name	financial_report.docx
Original SHA-256 Hash	
Original MD5 Hash	
Recalculated SHA-256 Hash	
Recalculated MD5 Hash	
Integrity Notes	The integrity of the file is verified as the hashes match; however, decryption was not attempted due to the focus being on the *client.xlsx* file. However, if there were more time, further attempts would be made.



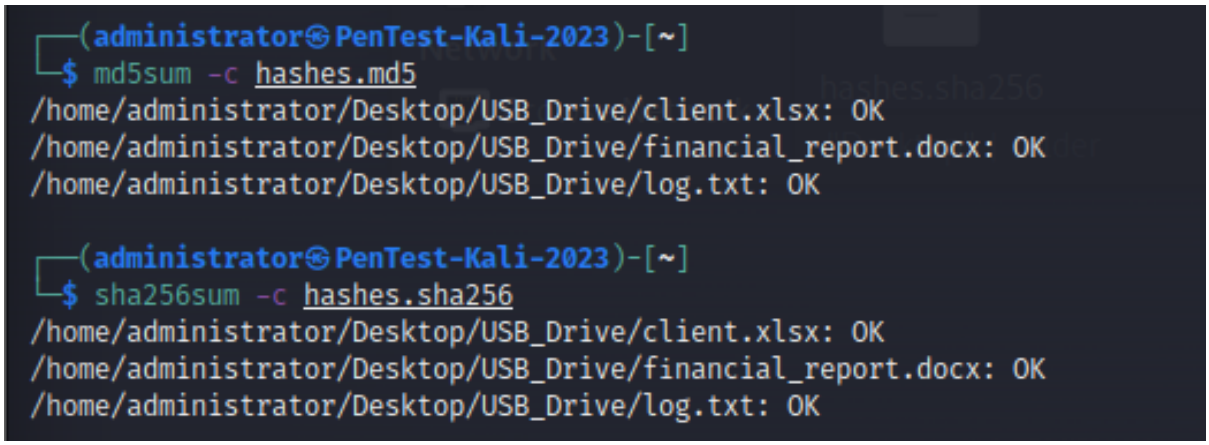
-- Image 3: Verification of file encyption.



-- Image 4: MD5 hashes generated for entire USB file.



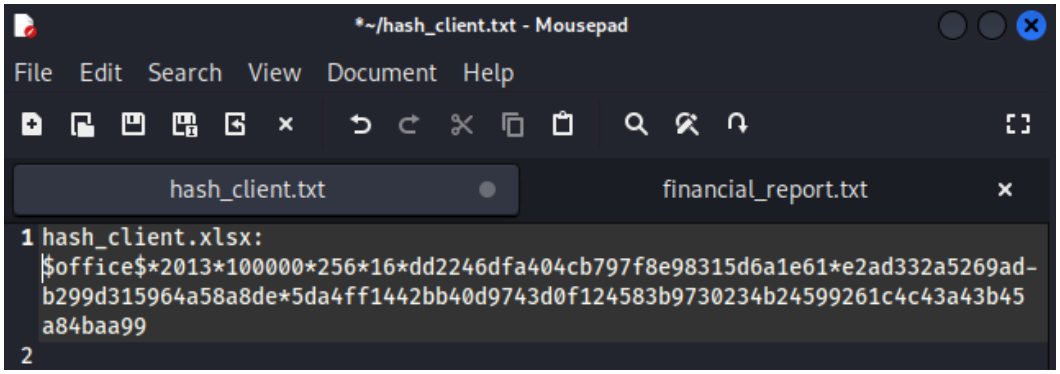
-- Image 5: SHA-256 hashes generated for entire USB file.



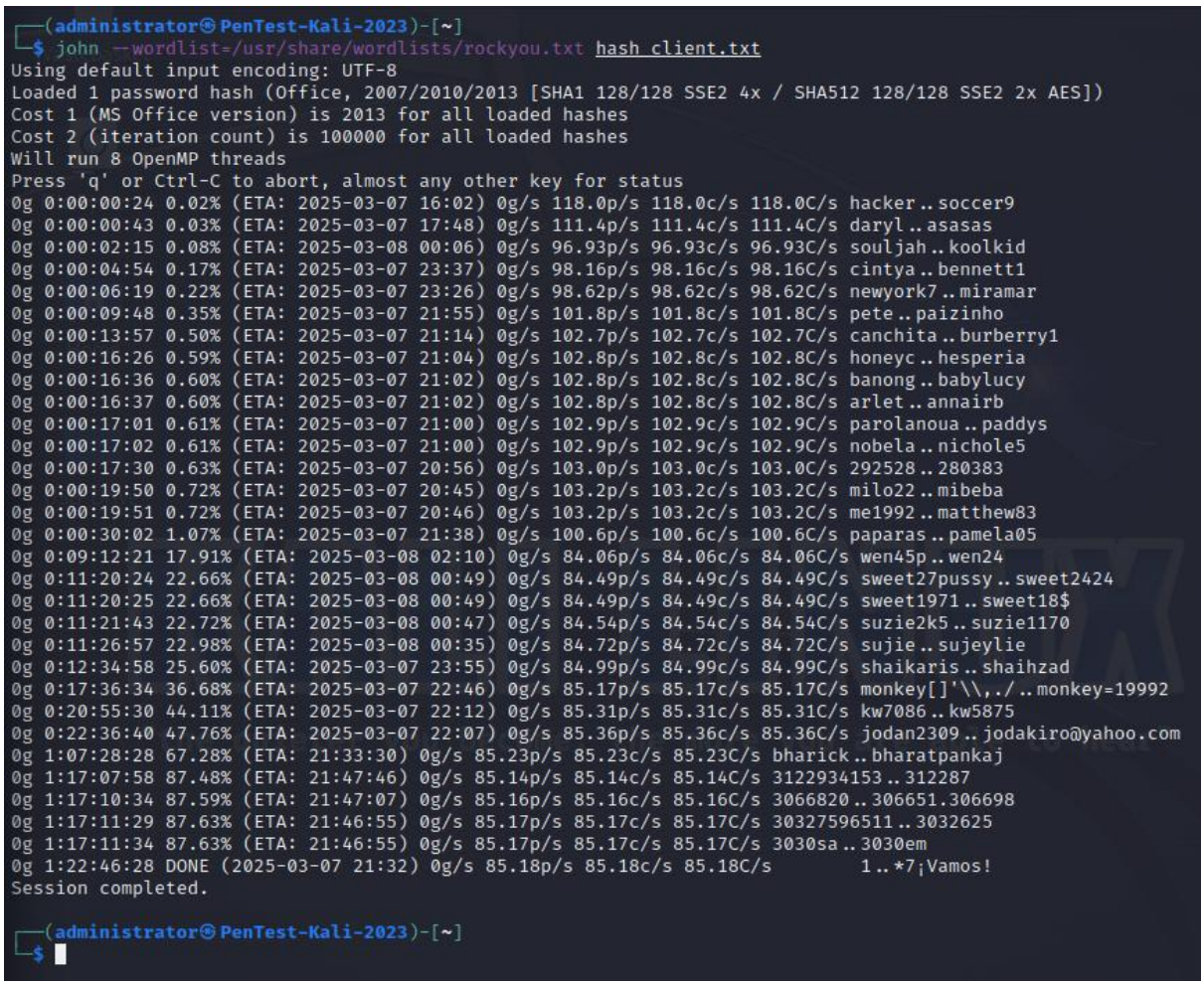
-- Image 6: Verification of matching hashes when stored as a text file for entire USB file.

3. Screenshots

-- Image 7: Extracts the hash file and saves it onto a text file for password processing tools.



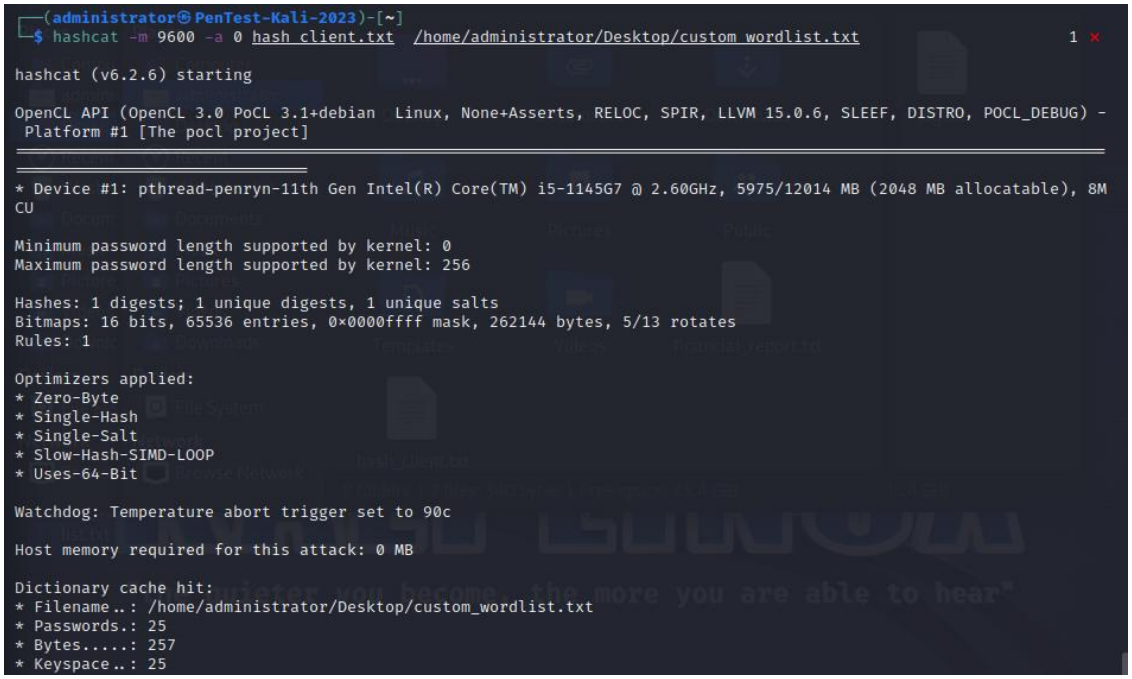
-- Image 8: Extracted hash text file for password processing.



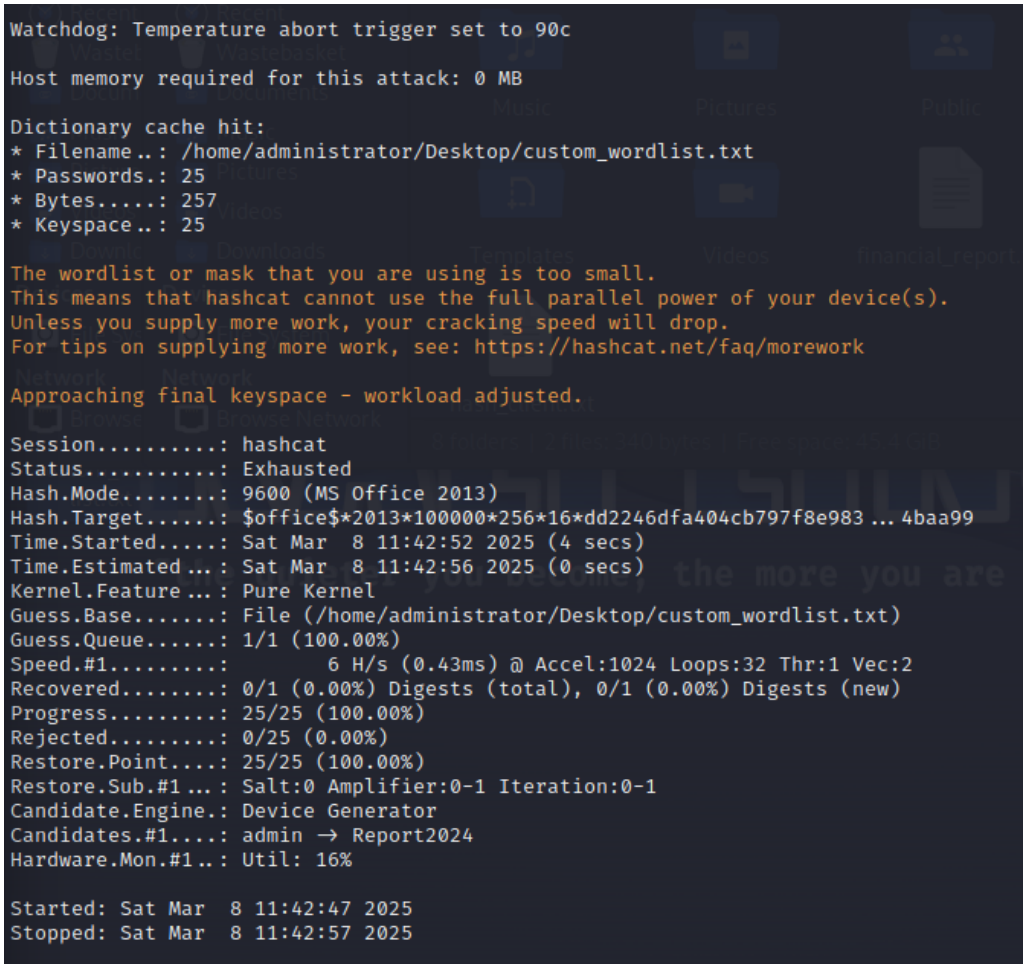
-- Image 9: The process of trying to crack the client.xlsx file using image 8 with John the Ripper – A password-cracking tool that was unsuccessful after a total time of 1 day and 22 hours.

```
admin
jean
alison
client_list
financial_report
budget2024
client2024
financial2024
20240201
01022024
192168110
192168115
192168120
192168125
192168130
192168135
192168140
192168145
192168150
password2024
admin2024
jean2024
alison2024
Finance2024
Report2024
```

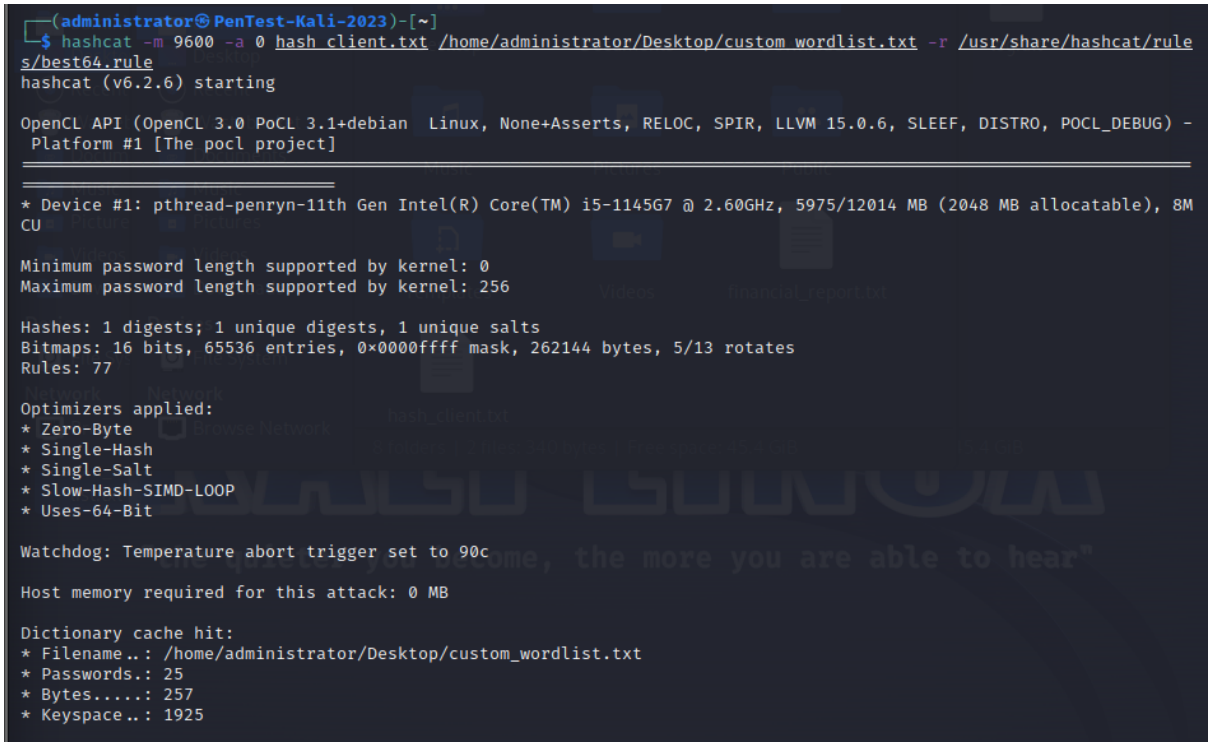
-- Image 10: Custom-created wordlist using the log file for any details on possible passwords.



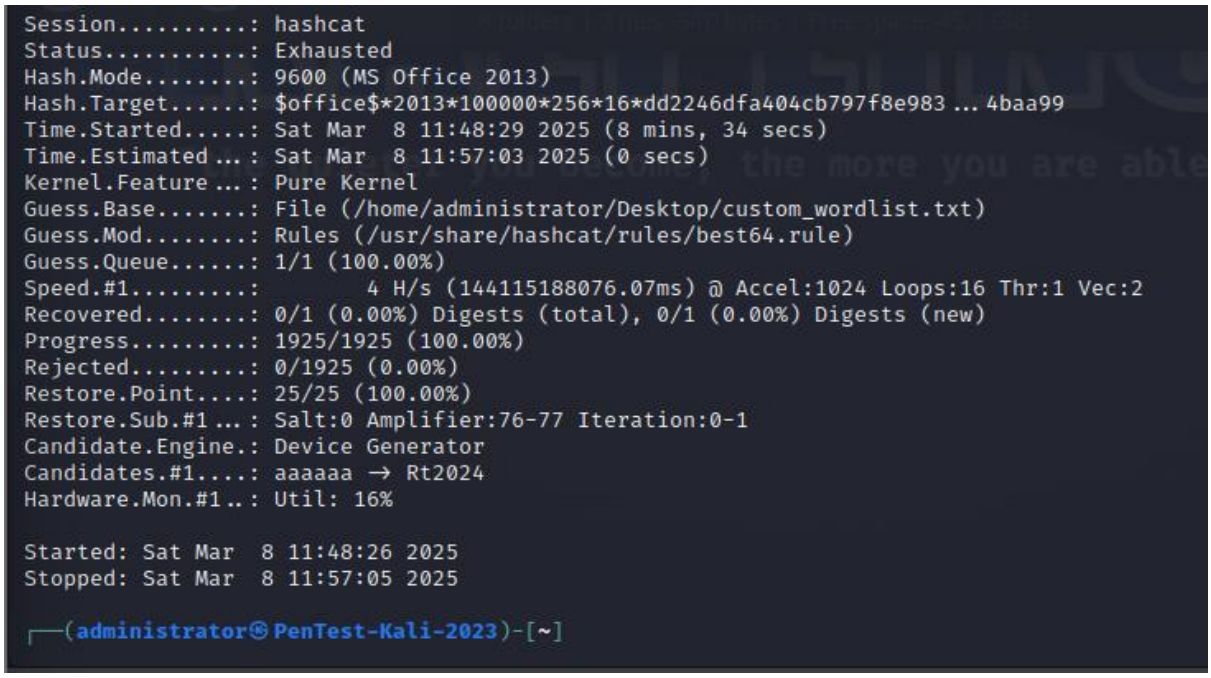
-- Image 11: Using hashcat (password cracking tool) to process custom wordlist. Note: "-m 9600" indicates the hash type, which is a 2013 office encrypted file and "-a 0" specifies a dictionary/wordlist attack.



-- Image 12: Hashcat failed to crack with custom wordlist.

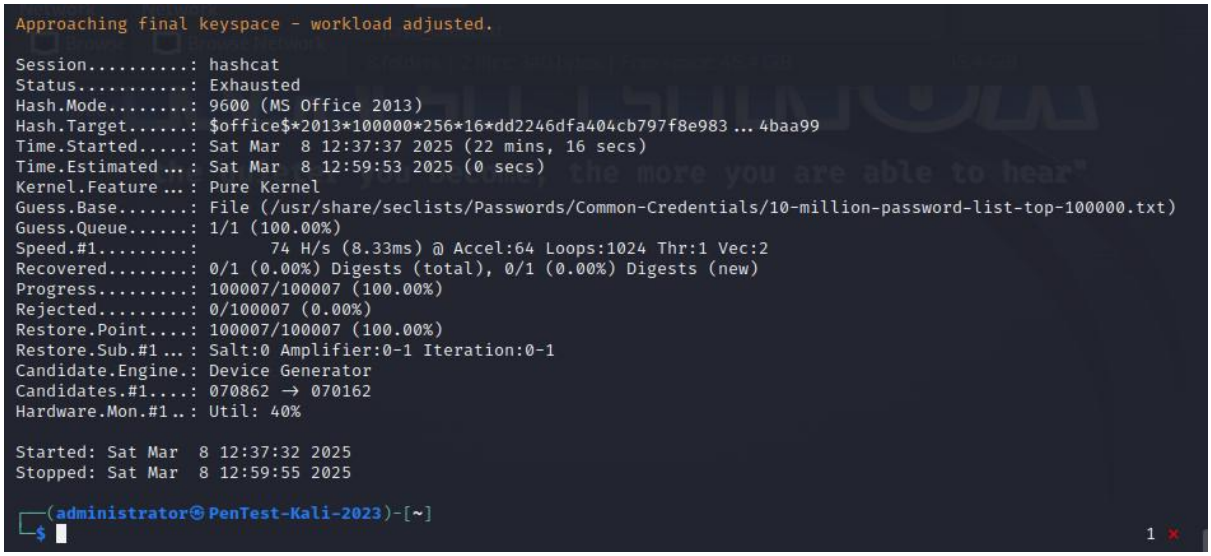


-- Image 13: Running hashcat with custom wordlist along with rule 64, which automatically applies common mutations to passwords within the wordlist generating multiple variations. Note: "-m 9600" indicates the hash type, which is a 2013 office encrypted file and "-a 0" specifies a dictionary/wordlist attack.

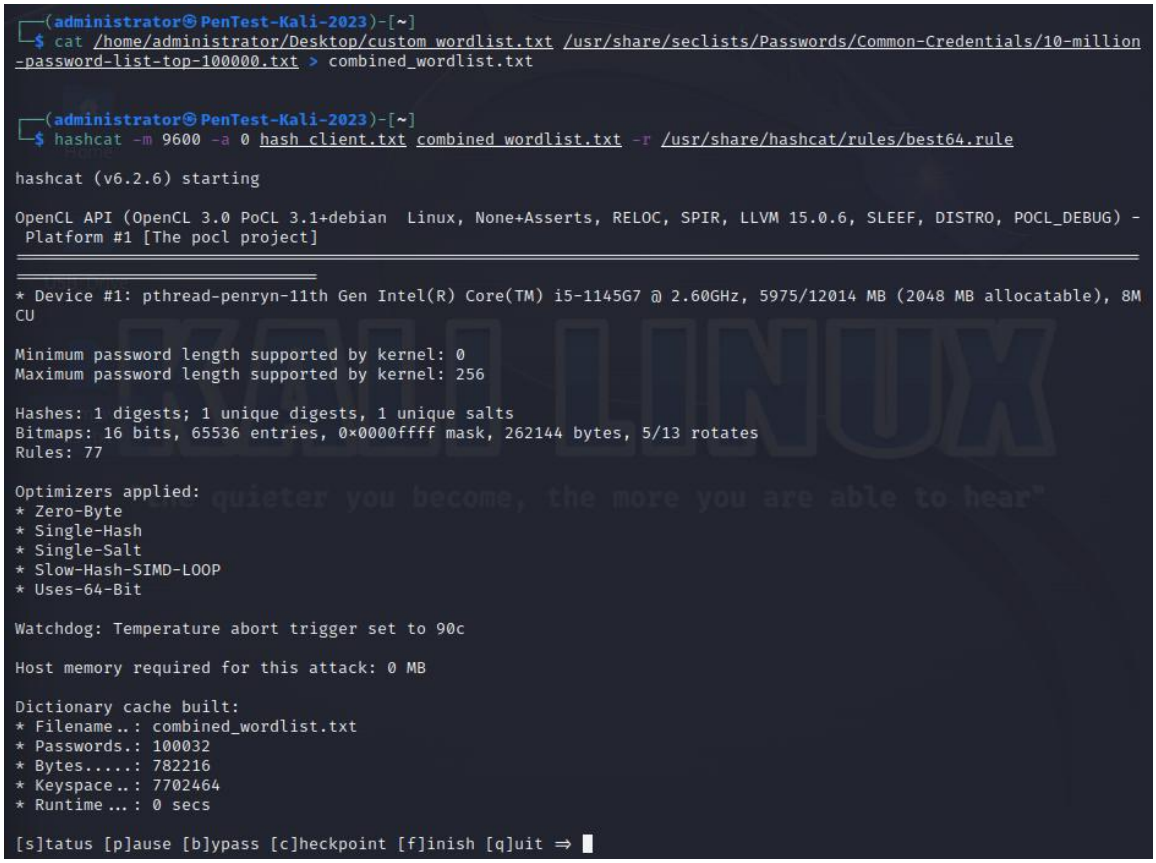


-- Image 14: Wordlist along with rule 64 was not able to decrypt the password on hashcat.

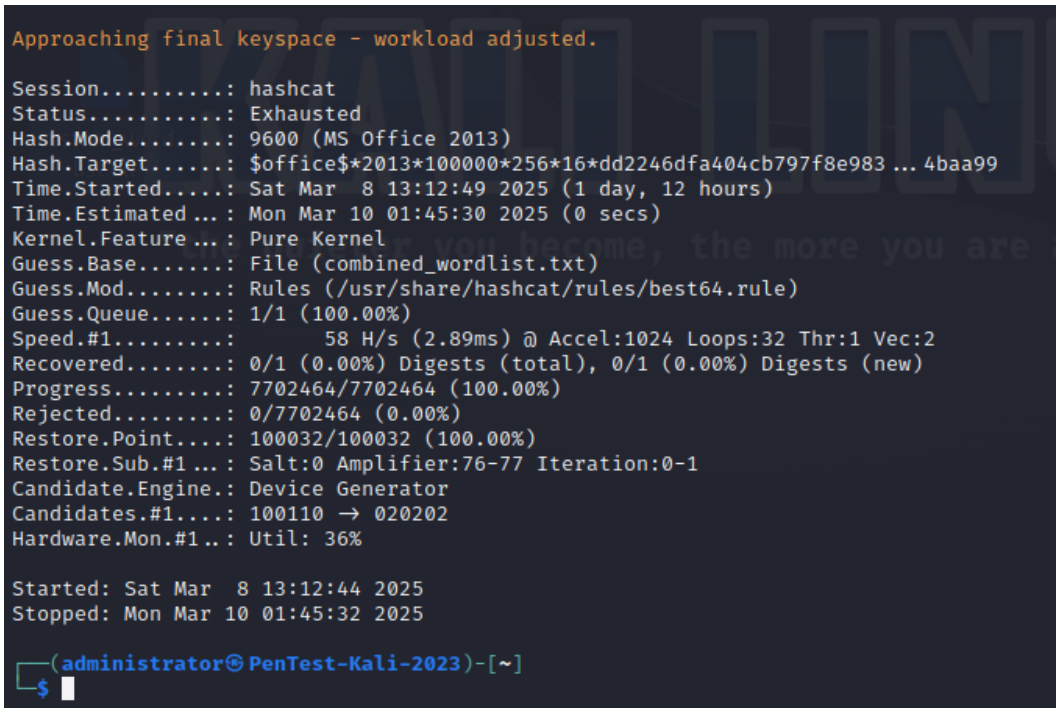
-- Image 15: Running hashcat with another extensive word list, including the 100,000 most common passwords. Note: "-m 9600" indicates the hash type, which is a 2013 office encrypted file and "-a 0" specifies a dictionary/wordlist attack.



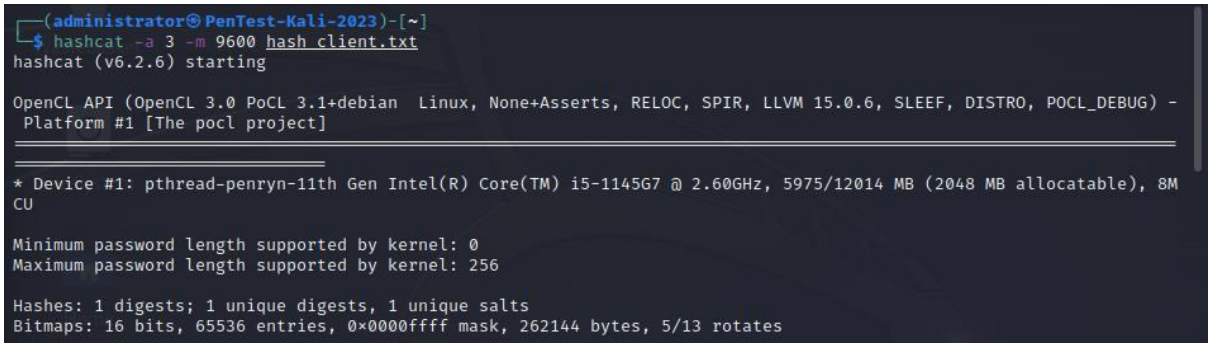
-- Image 16: Worlist was unable to decrypt the password.



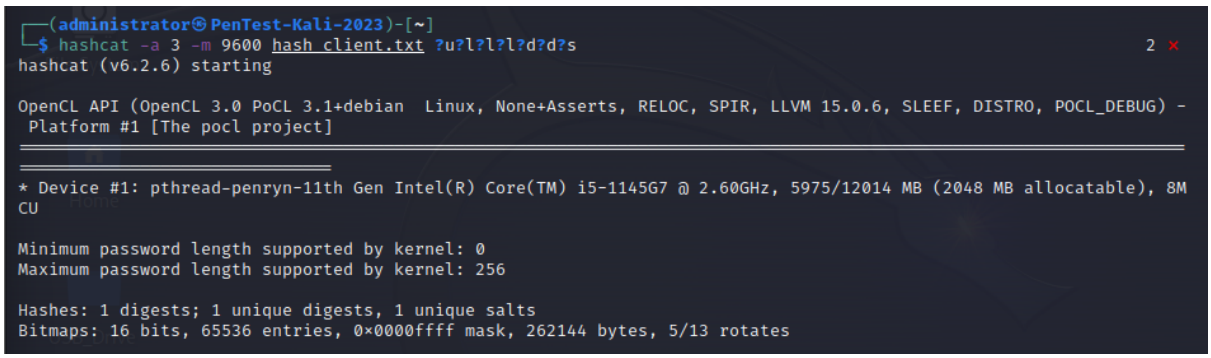
-- Image 17: Saving the top 100,000 password list and applying rule 64, which automatically applies common mutations to passwords within the wordlist, generating multiple variations. Note: "-m 9600" indicates the hash type, which is a 2013 office encrypted file and "-a 0" specifies a dictionary/wordlist attack.



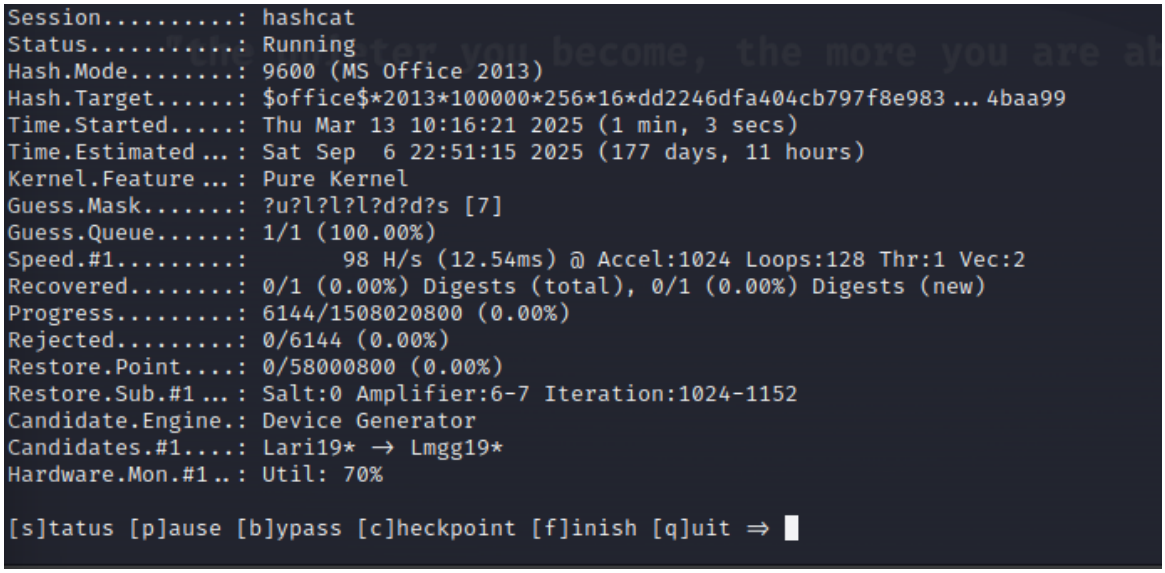
-- Image 18: The 100,000 password list and applying rule 64 was unable to decrypt the password after a total time of 1 day and 22 hours.



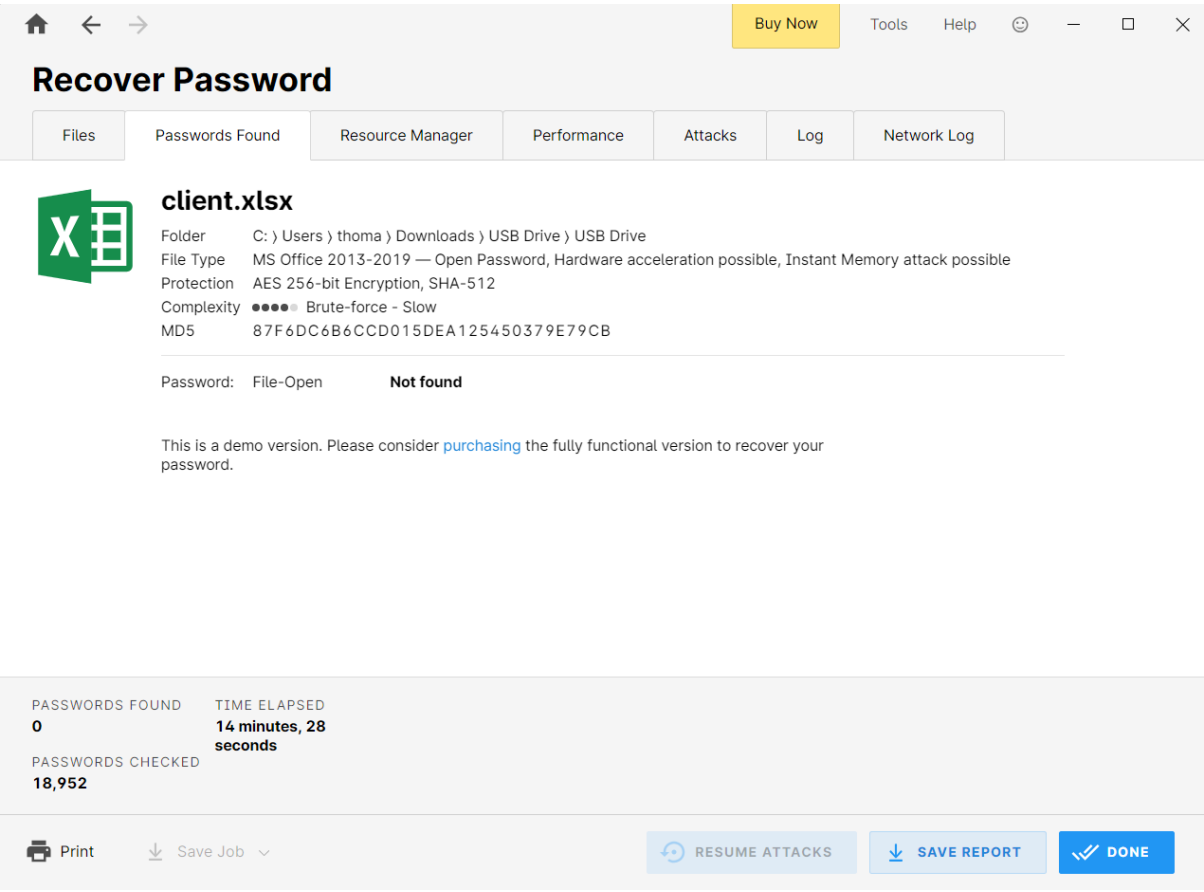
-- Image 19: Brute force is the only local next step; however, due to the limited time, it is not possible to decrypt. Note: "-a 3" indicates a brute force attack, and "-m 9600" indicates the hash type, which is a 2013 Office encrypted file.



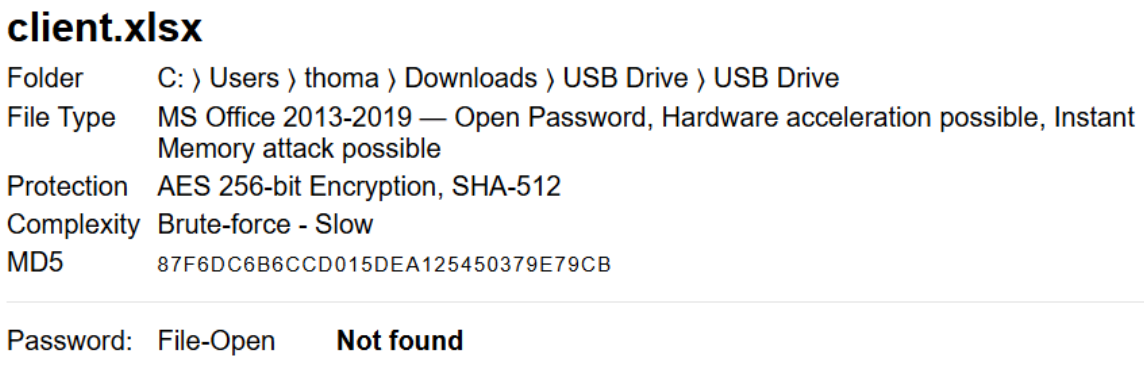
-- Image 20: Although there is no information available about the password's format, a typical password format theme was used; however, due to the nature of brute force, it was unable to complete due to the limited time. Note: "-a 3" indicates a brute force attack, and "-m 9600" indicates the hash type, which is a 2013 Office encrypted file.



-- Image 20: Shows the estimated time using a brute force method to decrypt the password.



-- Image 21: Attempt at using 'Passware Forensic Kit' software; however, no result was found for the password.



-- Image 22: 'Passware Forensic Kit' software results

4. Forensic Analysis

During this forensic investigation, the biggest challenge that was encountered was decrypting the encrypted file, which in this case was *client.xlsx,* which was encrypted with AES-256 encrption, which is well known for high ingerity and security against cryptographic based attacks. Targeted wordlists were used with well-known and extensive common passwords such as *rockyou.txt* and *10-million-password-list-top-100000,* which both failed to return any indication of decryption.

Advanced techniques such as Hashcat's Rule 64 were also applied to increase the possibility of further identifying the password if the structure was more complex. Despite creating a custom wordlist using the *log* file for analysis, this approach also failed. Further, multiple attempts using hashcat were also performed with rule-based modifications; many attempts took over two days to complete, but none succeeded. A final attempt was made using the software tool '*Passware Forensic Kit',* which is a powerful forensic tool that is designed to decrypt complex passwords; however, similarly, it failed to display any successful result.

The consistent failures of all of these methodologies to decrypt the file strongly suggest that the password is not included within a wordlist or dictionary-based and is more likely a sophisticated combination of characters with no pattern or lengths and are completely randomised. Brute force attack ETA time, even with full GPU utilisation, is impractical at 177 days.

In reflection, this forensic analysis highlighted the critical challenges with modern-day advanced encryption. The complexity of the encryption highlights the importance of implementing strict cybersecurity measures, such as enforcing strict password polices and maintaining detailed logs.

References

Mozilla Foundation, 2025. *403 Forbidden.* [Online]
Available at: https://developer.mozilla.org/en-US/docs/Web/HTTP/Status/403
[Accessed 12 03 2025].
NTNU, 2025. *Protect Microsoft Office documents with encryption.* [Online]
Available at: https://i.ntnu.no/wiki/-/wiki/English/Protect+Microsoft+Office+documents+with+encryption#:~:text=the%20message%20log.-,How%20secure%20is%20the%20encryption?,how%20to%20make%20secure%20passwords.
[Accessed 13 03 2025].
XPressGuards, 2025. *Decrypting Digital Evidence: Cyber Investigations.* [Online]
Available at: https://xpressguards.com/decrypting-digital-evidence-cyber-investigations/
[Accessed 13 03 2025].