

Journal Scientific and Applied Research, Vol. 30, 2026
International Journal

ISSN 1314-6289 (Print)
ISSN 2815-4622 (Online)

Original Contribution

A PRACTICAL IMPLEMENTATION OF SOFTWARE TOOLS FOR CHECKING THE LEVEL OF SECURITY IN THE USE OF PASSWORDS

Petar K. Boyanov

Department of Communication and Computer Engineering and Security Technologies, Faculty of Technical Sciences, Konstantin Preslavsky University of Shumen, Shumen 9712, 115 Universitetska St., e-mail: petar.boyanov@shu.bg

Abstract: *In this scientific article a practical implementation of software tools for checking the level of security in the use of passwords is shown. The article explains how password-cracking tools use hash values in a controlled laboratory environment without internet access. It also presents ways to create and protect strong passwords.*

Key words: *John the Ripper, Password security, Password cracking, Password hashes, Kali Linux, ZIP archives, Wordlist attack.*

1. Introduction

Information security is important because computers store and share a lot of personal and business data. One way to protect this information is to use passwords. Some passwords are easy to guess and may create security risks. This study examines the main features of John the Ripper. The tool can be used to test password strength and identify weak passwords. It can work with different types of password hashes and password-protected archives [3,4,7].

John the Ripper attempts to recover passwords by testing words, numbers, and character combinations. Systems usually store password hashes instead of plain-text passwords. The program tests different passwords and compares their hashes with the given hash. If a match is found, the tool displays the corresponding password. One of the methods used by the program is dictionary-based testing. It uses a list of common words and passwords to check if a password is easy to guess [8,9,10,11]. Another method is to test different combinations of letters, numbers, and special characters. The time needed for this process depends on the password length and complexity. John the Ripper provides several modes for testing passwords in different ways. T Users can use an existing wordlist or apply rules that modify words by changing letters and adding numbers or special characters. This makes the program useful for testing different types of passwords [2,5,6,7].

This scientific study uses the tool for two tasks. The first task is to test the security of a newly created Kali Linux user account by using its password hash. The second task is to test a ZIP archive that is protected with a password. The goal is to find the correct password and access the files stored inside the archive.

However, the tool cannot recover every password. Its success depends on several things, such as the password length, its complexity and the hashing method that is used. Simple passwords are usually easier to find, while long and complex passwords can take much more time to crack [1,2,3,4,7]. This tool is useful for learning more about computer security and understanding why strong passwords are important. Users can learn how weak passwords put their information at risk and why strong passwords are important.

The content of this academic article is intended for research, instructional applications and educational purposes only. Any unauthorized or unethical use of the presented material falls outside the author's scope of responsibility.

2. Related work

Previous research has examined password security and methods for identifying weak passwords. Agboluaje and Odejobi [1] compare John the Ripper with other tools for recovering passwords from protected files and archives. Other studies examine password testing methods based on words, letters, numbers, and symbols [2,4,11]. Several authors describe the operation of John the Ripper and its use in password security testing [3,5,6,10]. Other studies discuss password protection methods and compare tools used for password recovery [7,8,9].

3. Problem description and results

The experiments in a controlled laboratory environment were carried out. The used operating system is Kali linux x64 with the following system information: Linux Petar 7.1.5+kali-amd64 #1 SMP PREEMPT_DYNAMIC Kali 7.1.5-1kali1 (2026-07-29) x86_64 GNU/Linux.

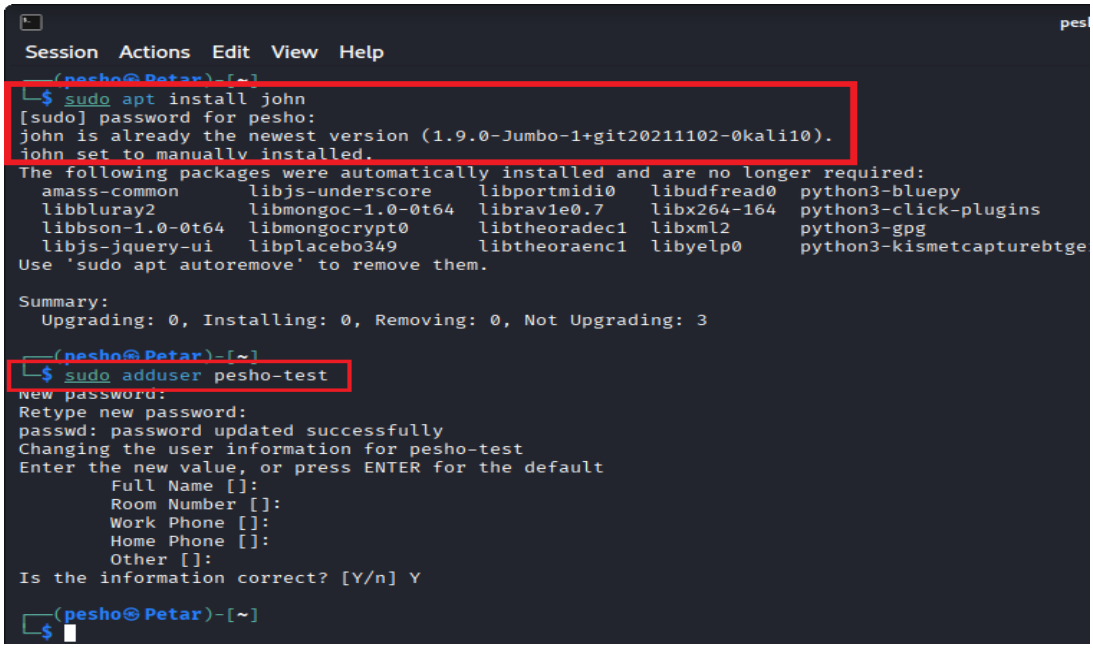


Fig. 1. Installation of the software tool and the creation the first account

The first task was to check whether John the Ripper was installed on Kali Linux After running the appropriate terminal command, the system showed that John the Ripper was already installed. The installed version was 1.9.0-jumbo-1+bleeding-5184bc0 2026-10-11 14:18:59 +0200 OMP [linux-gnu 64-bit x86_64 AVX512F AC]. Two new user accounts were created for the experiment. Their usernames were "pesho-test" and "pesho-test-account". The

password for both accounts is 8 characters long, namely "12345678". This password is widely used and is also very easy to guess [2,3,4,6,7,11]. This is shown in fig. 1 and 2. Fig. 2 shows a successful login to the second test account.

The next step was to display the password hashes of both accounts. This is done by typing the command "cat /etc/shadow" [1,2,3,5,8,9,10]. Both hashes were copied into a text file. John the Ripper then used this file to test possible passwords for the two user accounts.

The hash function of the first user is "pesho-test:yj9T$9ALERvxC41bvIuN7mD6DN1$XJdtIEtzm5oWp0UycXFKKyCb3UxiIjPLq/LODuutUW4:20757:0:99999:7:::" and the hash of the second user is "pesho-test-account:yj9T$PThroRoJeEBu/JcQFT1Da.$GNTpVRjTrmaEX7gK4KhBuslJr5Kx/OudzbNZXxeDEO4:20757:0:99999:7:::". The hashes in fig. 3 are presented.

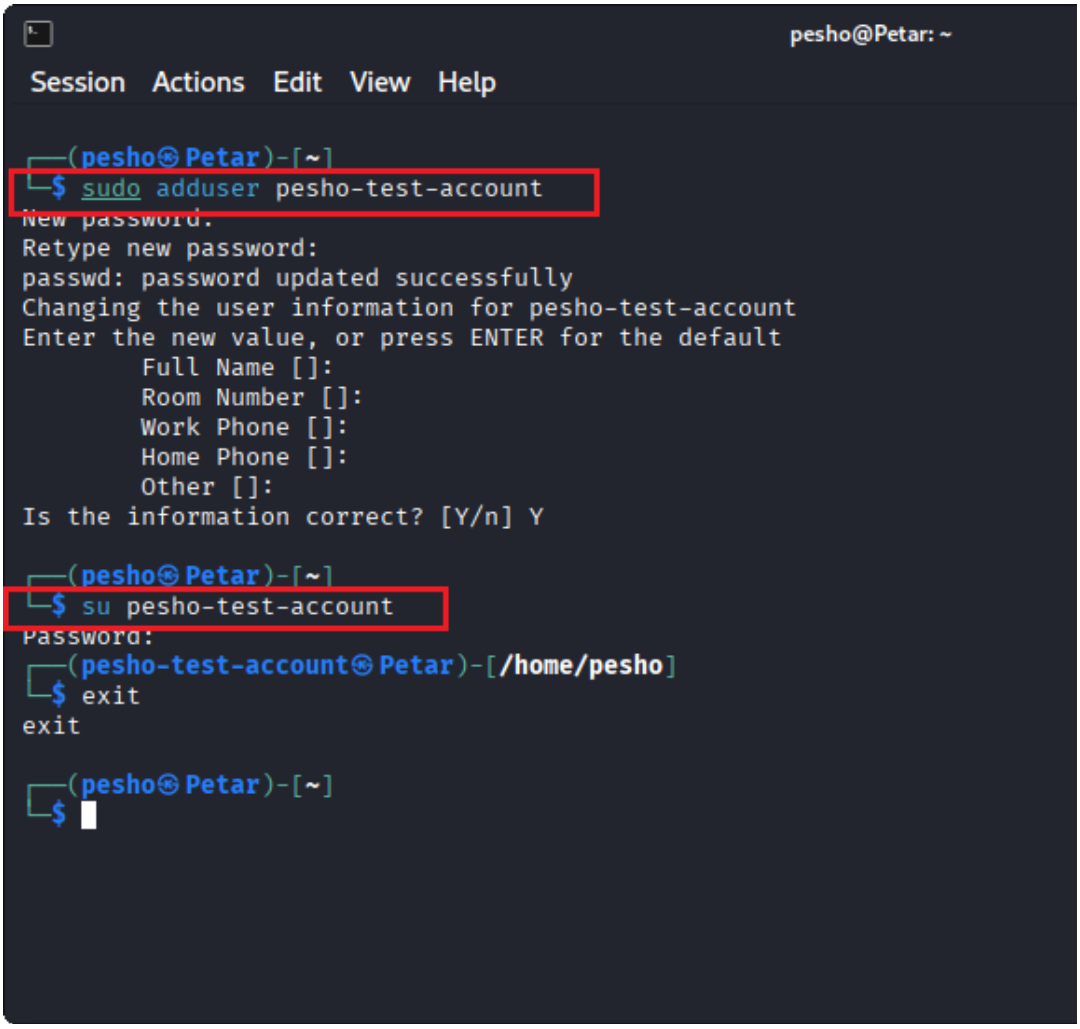


Fig. 2. The process of creation of the second account

John the Ripper supports several password-hashing formats, including SHA512, SHA256, Bcrypt, Yescrypt and MD5 [1,3,4,6,9,10,11]. John the Ripper uses a wordlist to test possible passwords against the stored hashes. When a matching password is found, the tool displays it in plain text. For the purposes of scientific research, the successful disclosure of the password "12345678" of the second account - pesho-test-account - is shown (Fig. 5).

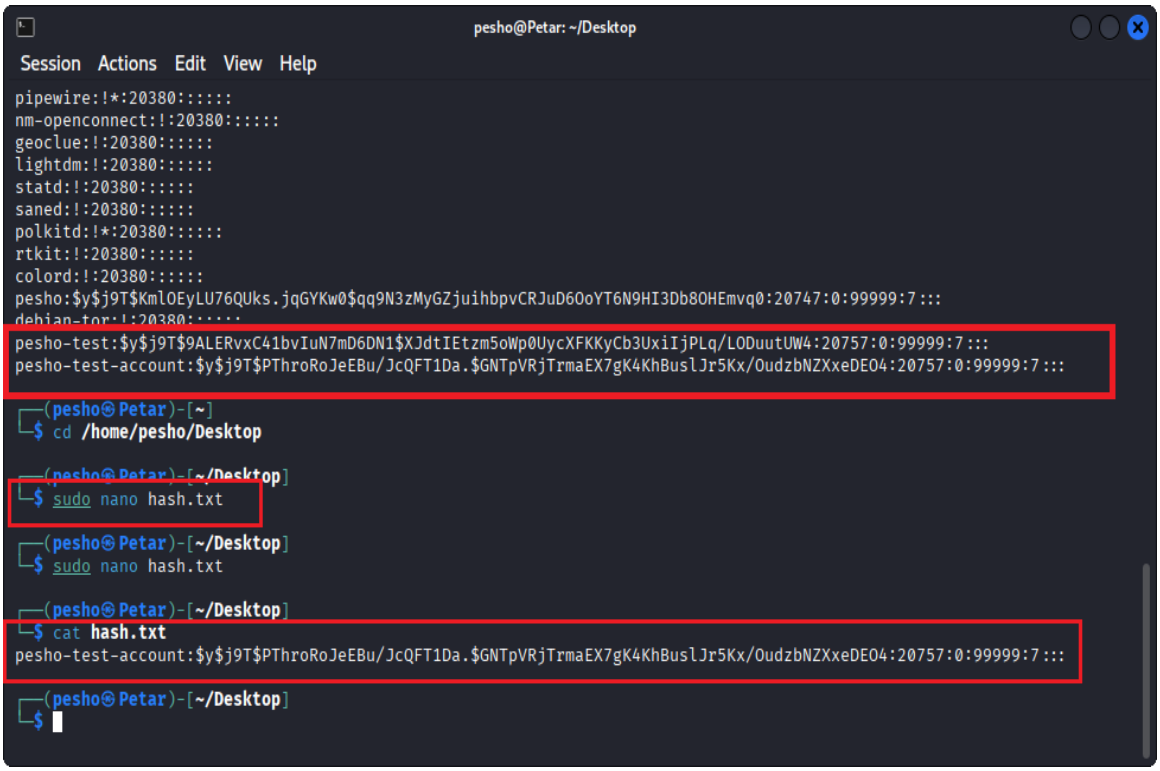


Fig. 3. The hashes of both accounts

The second task was to create a text file containing the message "Test information for the research!" (Fig. 4). The file was then compressed into a password-protected ZIP archive using the password "12345678".

Fig. 7 shows several unsuccessful attempts to open the archive. After the third attempt, the system displays an incorrect-password message. Protecting backup files with passwords can help prevent unauthorized access to confidential information. When there is a password on the backup file, then even no antivirus program can scan the file because it cannot reveal the password. If an unprotected archive contains an executable file, an antivirus program may detect it as potentially harmful.

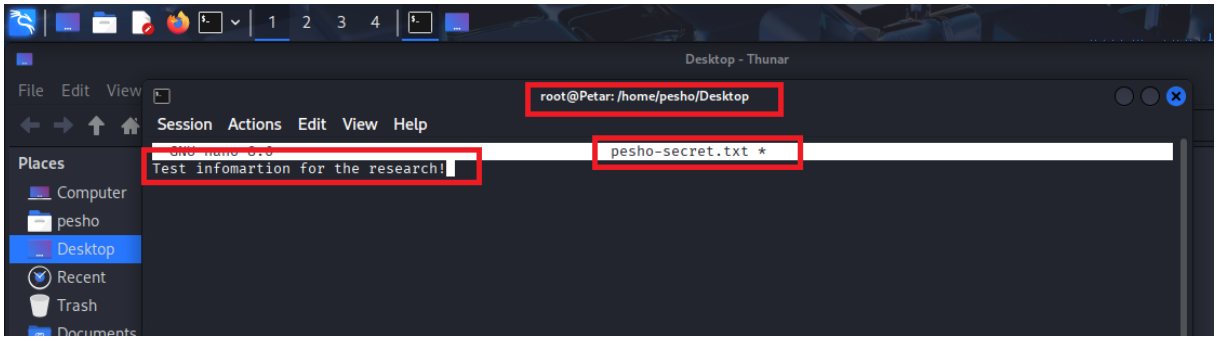


Fig. 4. The process of creation of the plain text message

The next step was to prepare a list of test passwords, including the correct password "12345678". The passwords are respectively (fig. 6):

- „123“;
- „1234“;
- „12345“;
- „123456“;

- „12345678“;
- „123456789“;
- „12345678910“.

The test list can be expanded with many combinations of numbers, lowercase and uppercase letters, and special characters.

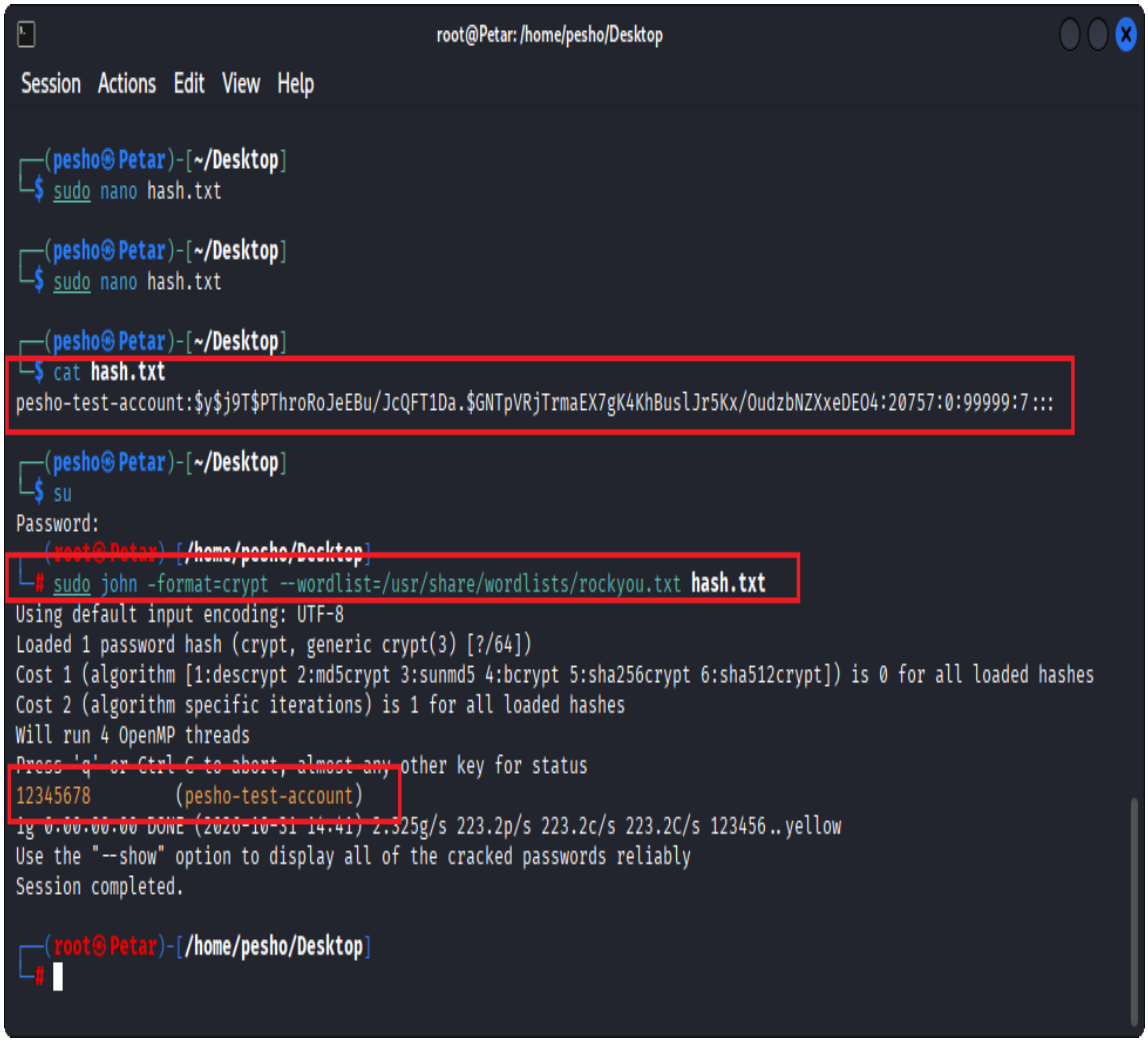


Fig. 5. The revealed password of the second account



Fig. 6. The process of entering random passwords

A UTF-8 encoded text file was created using the nano editor. The editor runs in the terminal and is controlled through commands. To complete the second task, the built-in tool "zip2john" was used. The tool extracts the password hash from a ZIP archive.

After running the tool, the extracted password hash was displayed in the terminal - „pesho-secret.zip/pesho-secret.txt:$pkzip$1*2*2*0*2f*23*fb7194f1*0*4a*0*2f*762e*22bf555ca154b17a1fc1cd75474b1f94e07471abbf406dcb84796460942de85873298c962b7e553486cfceb518f878*$/pkzip$:pesho-secret.txt:pesho-secret.zip::pesho-secret.zip". The extracted hash was displayed using the cat command. This is shown in fig. 8.

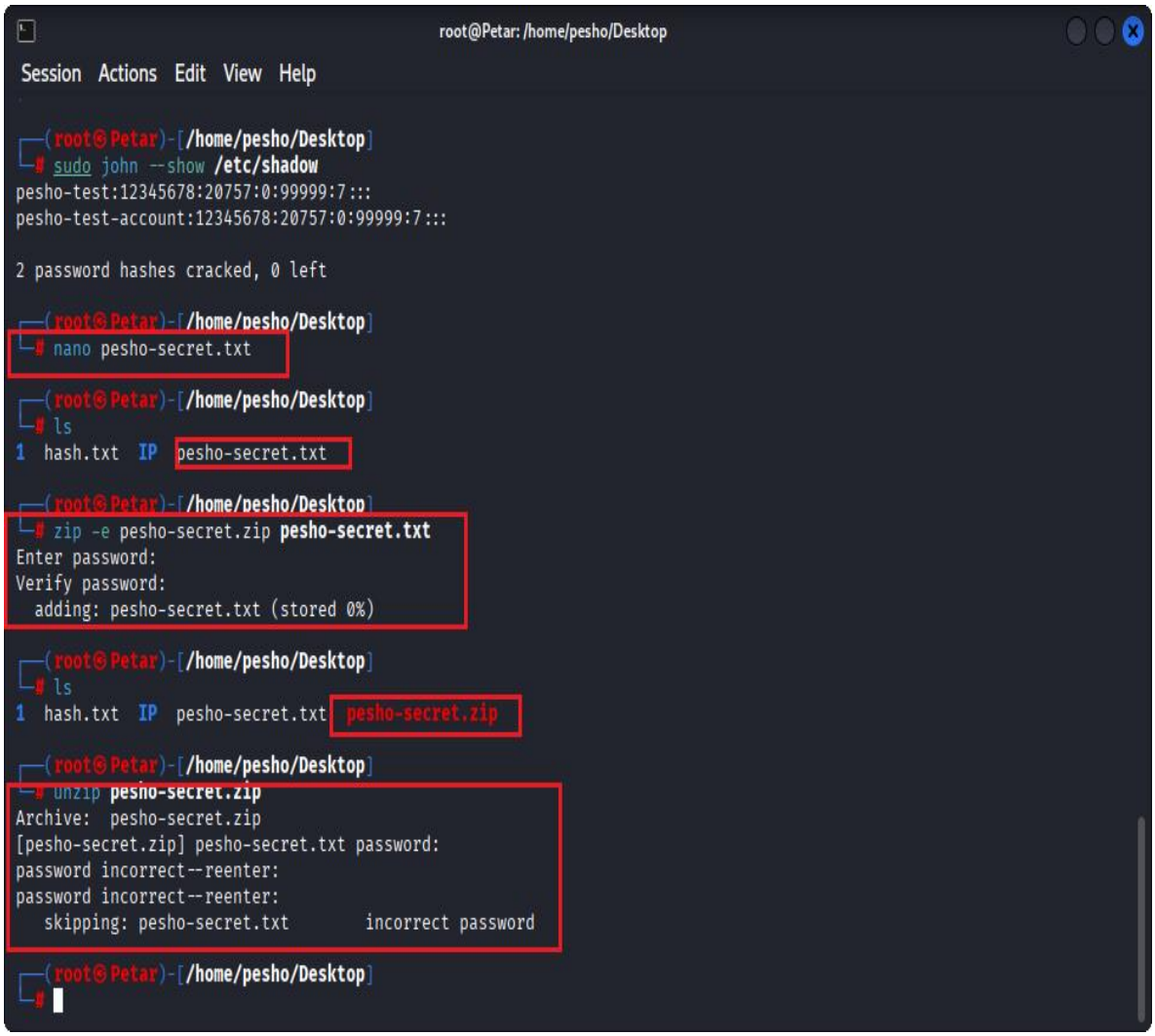


Fig. 7. The creation of password for the zip file

The final step was to recover the password of the ZIP archive. The following command was used: "sudo john --wordlist=pesho-passwords.txt zip.hash". John the Ripper recovered the password "12345678", as shown in Fig. 8.

Fig. 9 shows how the ZIP archive was extracted using the recovered password. The contents of the text file are also shown.

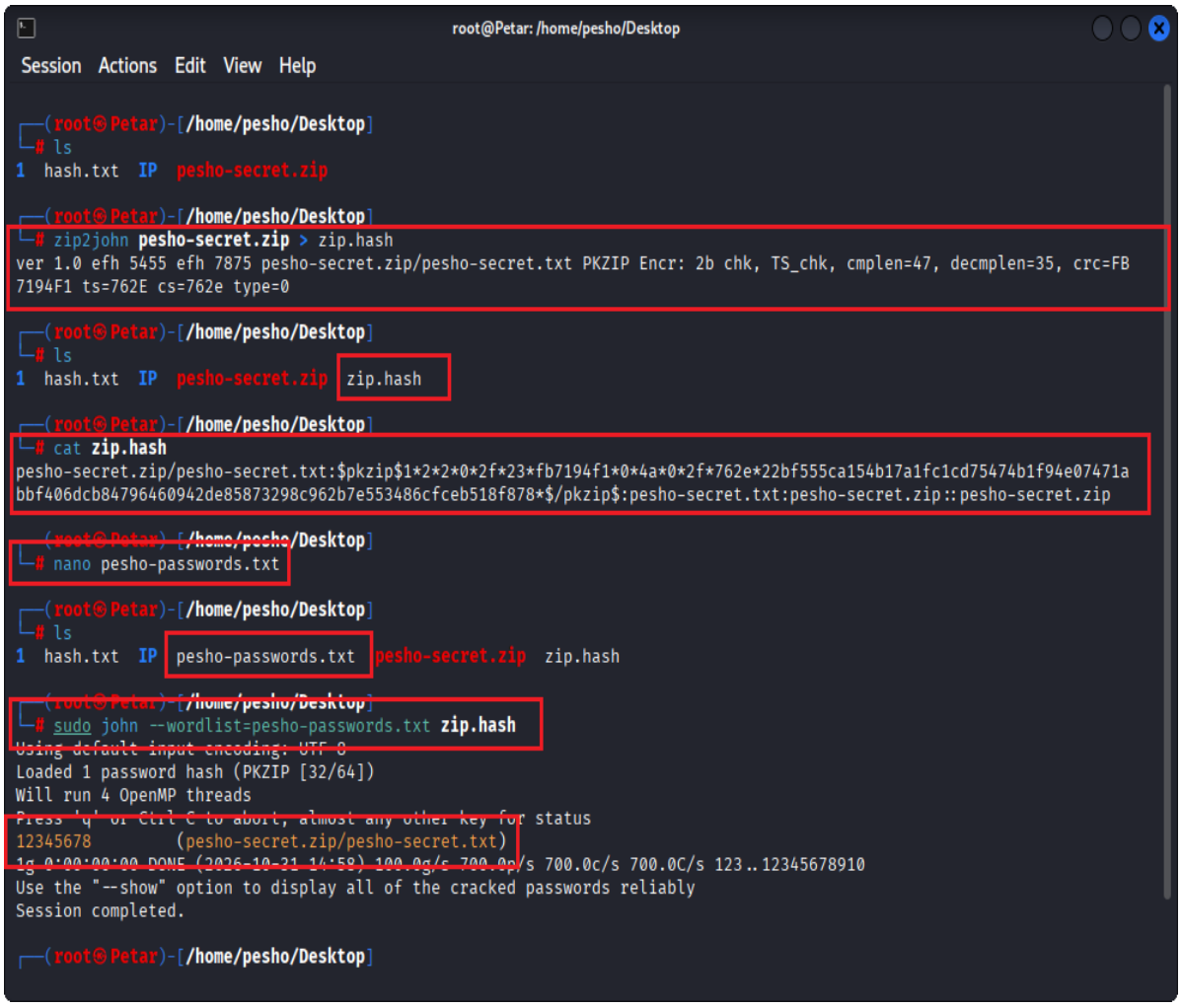


Fig. 8. The revealed password of the zip file

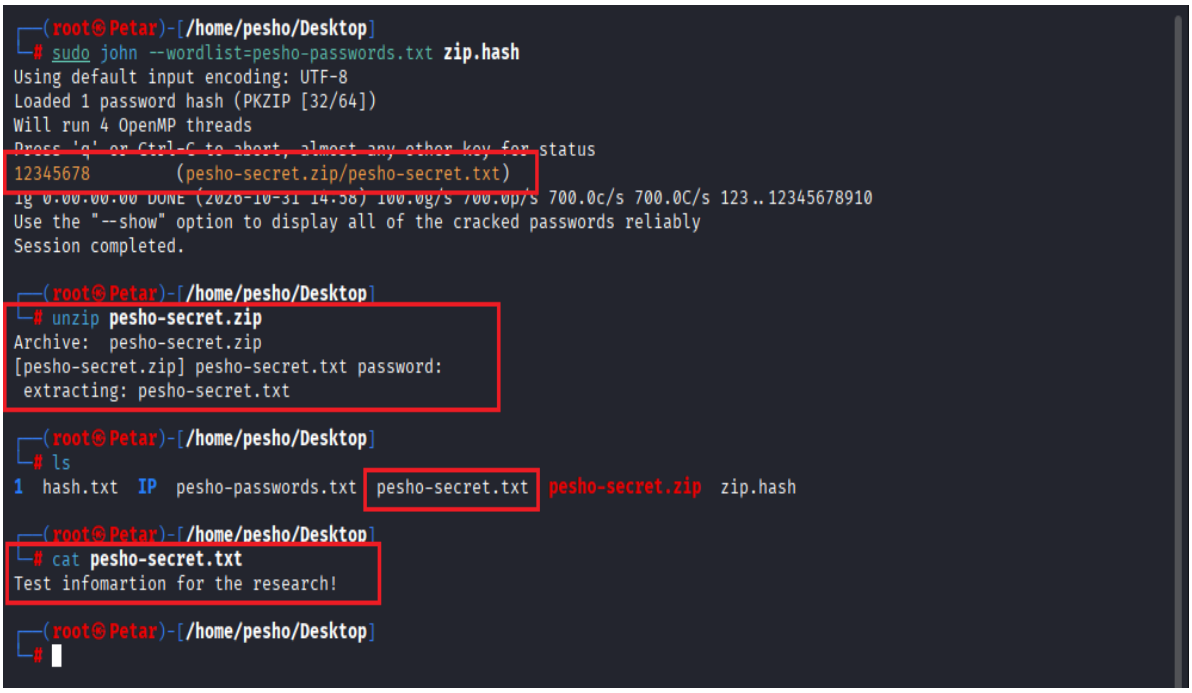


Fig. 9. The information inside the file

4. Conclusion

Weak passwords increase the risk of unauthorized access to confidential information stored on computers and servers. Users should choose long passwords that include a combination of letters, numbers, and special characters. Users should also avoid passwords based on personal information, such as names or car registration numbers, as well as default passwords. Using passphrases made up of several words is another way to improve password security.

References

[1] Agboluaje, R., & Odejobi, O. (2024). Comparative Analysis of John the Ripper, Cain and Abel, and Hashcat for Password Recovery in Encrypted Files and Archives.

[2] Di Campi, A. M., Focardi, R., & Luccio, F. L. (2022, September). The revenge of password crackers: Automated training of password cracking tools. In European symposium on research in computer security (pp. 317-336). Cham: Springer Nature Switzerland.

[3] Leyte-Vidal, J. (2024). Ethical Password Cracking: Decode passwords using John the Ripper, hashcat, and advanced methods for password breaking. Packt Publishing Ltd., ISBN 1804613851, 9781804613856, p. 168.

[4] Liu, E., Nakanishi, A., Golla, M., Cash, D., & Ur, B. (2019, May). Reasoning analytically about password-cracking software. 2019 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 2019, pp. 380-397, DOI: 10.1109/SP.2019.00070.

[5] Lubeck, T. (2013). Distributed Password Cracking with John the Ripper. Computer Security–Tufts Comp116,

[6] Marchetti, K., & Bodily, P. (2022, May). John the Ripper: An examination and analysis of the popular hash cracking algorithm. 2022 Intermountain Engineering, Technology and Computing (IETC), Orem, UT, USA, 2022, pp. 1-6, DOI: 10.1109/IETC54973.2022.9796671.

[7] O'Leary, M. (2026). User Authentication and Passwords. In Linux Security Foundations (pp. 113-172). Apress, Berkeley, CA, Online ISBN 979-8-8688-2664-1, https://doi.org/10.1007/979-8-8688-2664-1_3.

[8] Pahuja, D., & Sidana, P. (2021). Implementing and comparing different password cracking tools. International Research Journal of Engineering and Technology (IRJET), 8(5), 2089-2095.

[9] Shi, C., Li, Z., & Li, X. (2025). System Password Security: Attack and Defense Mechanisms. arXiv preprint arXiv:2510.10246, https://doi.org/10.48550/arXiv.2510.10246.

[10] Sykes, E. R., Lin, M., & Skoczen, W. (2010, November). MPI enhancements in John the Ripper. In Journal of Physics, Volume 256, High Performance Computing Symposium (HPCS2010) 5–9 June 2010, Victoria College, University of Toronto, Canada, DOI: 10.1088/1742-6596/256/1/012024.

[11] Weir, M., Aggarwal, S., De Medeiros, B., & Glodek, B. (2009, May). Password cracking using probabilistic context-free grammars. 2009 30th IEEE Symposium on Security and Privacy, Oakland, CA, USA, 2009, pp. 391-405, DOI: 10.1109/SP.2009.8.