WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Ransomware Prevention Through Backup Automation and Secure Storage Strategies in Windows

Parthiv Dudhat[1], Hetshree Sathwara[2], Jayvin Gohel[3], Krisha Sidhdhapuria[4], Purvi Baria[5], Aditya More[6], Dr. Kapil Kumar[7]

[1,2,3,4,5-] Student, Integrated M.Sc. Cyber Security and Forensics, Department of Biochemistry and Forensic Science, Gujarat University, Ahmedabad, Gujarat- INDIA

[6-] Teaching and Research Associate, Cyber Security and Forensics, Department of Biochemistry and Forensic Science, Gujarat University, Ahmedabad, Gujarat- INDIA

[7-] Co-ordinator and Associate Professor, Department of Biochemistry and Forensic Science, Gujarat University, Ahmedabad, Gujarat- INDIA

ABSTRACT-

The study presents the development of the tool aimed at defending user data against ransomware attacks by automating secure backup and recovery processes. This Python-based tool enhances data protection through strategic storage and file format transformation. This study effectively reduces the risk of ransomware encryption by converting critical sensitive files to less targeted formats and relocating them to secure system directories. Comprehensive testing against the ransomware samples demonstrated the tool's reliability in protecting the data and enabling seamless recovery. This work strengthens ransomware defence strategies, ensuring robust data security and resilience for end users.

List of keywords: Ransomware protection, auto backup, encryption attack, ransomware recovery, data security.

I. INTRODUCTION

One of the most significant challenges in the digital world currently is ransomware. It is becoming a major danger and issue for businesses, organisations, healthcare systems, and information security specialists [1]. Ransomware acts as a form of extortion which encrypts the files in a victim's computer and forces the victim to pay the ransom to have the data recovered [2]. While large organizations often employ robust defences and backup strategies, individual users, particularly non-technical ones such as students or casual users, are frequently left vulnerable. This demographic often lacks awareness of effective backup practices or how to safeguard personal and work-related data, making them prime targets for ransomware attacks. The introduction of this paper presents a novel tool designed to address this disparity by enhancing data security measures at the end-user level. Focusing on the protection of critical user files and data, the tool operates on the principle of modifying file formats and storing data in locations less susceptible to ransomware encryption, thereby mitigating the impact of attacks on user devices and operating systems. This introduction establishes the foundation for exploring the development, implementation, and effectiveness of this specialized tool in combating ransomware and safeguarding the digital assets of vulnerable user groups.

II. RELATED WORK

In the current scenario of ransomware attacks, many methods have been presented. One such approach involves the use of Alternate Data Streams (ADS) in the Windows file system. ADS allows for the storage of additional metadata with files, which can be harnessed to create hidden, local backups that remain unnoticed by ransomware. By utilizing ADS, it is possible to develop a stealthy and efficient backup mechanism that protects user data without the need for constant remote data transfer [3].

Khan et al. introduced an innovative ransomware protection method utilizing the Moving Target Defense (MTD) approach to dynamically alter the attack surface, thereby reducing the attack success ratio. Their methodology employs multiple layers of MTD to enhance the security of user

files. The first layer generates random extensions that conceal the existing known file extensions, effectively safeguarding user files against ransomware variants that target specific file extensions.

The second layer of protection involves event-based MTD, where tasks are scheduled to change file extensions upon the occurrence of specific events, typically triggered by the execution of ransomware in the system. This multi-layered approach has demonstrated significant efficacy in protecting user files against several notorious ransomware variants, including WannaCry, Cerber, Locky, Tesla, Revil, Bitlocker, Darkside, and Ranzy. The experimental results indicated that the proposed method effectively mitigated ransomware attacks by making it more challenging for ransomware to identify and encrypt target files [4].

Lee et al. proposed a novel approach to protect critical files from ransomware by employing a hiding strategy that makes it difficult for attackers to locate the target files. This method involves the use of link files and an encrypted database to further enhance security and usability. By incorporating a linker and an encrypted database, their strategy effectively reduces the attack surface. The implementation of these best practices ensures that critical files are well-protected from existing ransomware threats while maintaining system usability [5].

III. PROPOSED METHODOLOGY

Most of the files ransomware targeted were in document storage formats like Office, PDF, and CSV [1]. To protect end-users critical and important data files from ransomware attacks, we propose a novel approach that involves changing the file format of the data to a Dynamic Link Library (.dll) format and placing it in a system directory typically ignored by ransomware. This method leverages the fact that ransomware often targets user directories and common file types for encryption, while system directories and files with .dll extensions are generally left unencrypted to ensure system stability.

A. RESEARCH DESIGN:

This research aims to address the critical need for user-friendly ransomware protection tools for non-technical users. The approach integrates robust technical components and systematic procedures to ensure effective data protection and recovery against ransomware threats. Key components include:

Destination folder for backups: "backup works"

Extraction tool: 7zip open source

Backup batch file: backup.bat

Tool run file: abcd.py

Recovery batch file: recovery.bat

B. SYSTEM CONFIGURATION:

The system requirements include Windows 10 or later, equipped with an Intel Core i3 processor or higher, at least 8GB of RAM, and sufficient storage space on the C-drive, depending on the size of the backup data. Essential software includes Python 3.x, 7-Zip, and batch scripting tools. Users should download the tool package from a trusted source and complete the installation on their Windows operating system. Additionally, the tool relies on Python libraries such as schedule, subprocess, time, random, and string for its operation.

C.USER REQUIREMENTS:

To set up the backup script, users need to specify the paths for the backup folder, temporary folder, and system32 directory, and provide the location of the 7-Zip executable file.

For the recovery script setup, users must specify the paths to the 'backup.dll', 'log.dll', temporary folder, and the location of the 7-Zip executable file.

D.BACKUP PROCESS:

In the initial phase of the backup process, we establish a consistent encryption key by embedding a static password within the 'abcd.py' script to ensure data security. When executed, 'abcd.py' starts the encryption process by dynamically generating a unique password for each backup session to improve security.

FIGURE 1. Backup Process

This randomly generated password, known as the dynamic password, is then passed to 'backup.bat'. Here, it undergoes hashing to securely manage and store it throughout the backup operation. Utilizing this hashed password, 'backup.bat' compresses critical files into a zip archive, effectively securing the data. As an added security measure, the resulting zip file is renamed with a .dll extension.

Additionally, the dynamic password is logged into 'log.txt' to facilitate recovery. This log file is also compressed into a zip archive using the static password for secure storage. Similar to the primary backup file, the 'log.txt' zip archive is renamed with a .dll extension. Both 'backup.dll' and 'log.dll' files are securely stored within the system32 directory. This directory was chosen due to ransomware's tendency to avoid altering files within system32, including .dll files. This comprehensive approach ensures our ransomware protection tool not only secures data efficiently but also maintains high security and integrity throughout the backup and storage process.

E. RECOVERY PROCESS:

The recovery process starts by running the 'recovery.bat' script. This script first converts 'log.dll' into 'log.7z' and 'backup.dll' into 'backup.7z', making sure the necessary files are ready for recovery. Next, the script will ask you to enter a static password used previously in 'abcd.py' during the backup process. This password is crucial for unpacking 'log.7z' and retrieving 'log.txt' from it, which contains crucial data needed for the recovery.

Once the 'log.txt' is accessed, the script retrieves a dynamic password from it. Subsequently, the script generates a hashed version of this dynamic password. Using this hashed password, the script proceeds to extract 'backup.7z', ensuring only authorized users with the correct credentials can access the backup data.

FIGURE 2. Recovery Process

Finally, after these steps, the backup data is successfully recovered. This comprehensive process ensures that data can be reliably retrieved, following strict security measures to prevent unauthorized access or data loss.

FIGURE 3. Malware Execution Flowchart

As observed in the above figure in the event of a ransomware intrusion, the tool may cease functioning, potentially halting the backup process. However, this does not affect the integrity of previously backed-up data, ensuring that recovery remains possible. Conversely, if the system remains unaffected by ransomware, the tool will continue its regular backup operations, safeguarding data as intended.

VI. RESULTS

Hitobito ransomware [first seen: 2024]

Obtain the latest Hitobito ransomware sample from MalwareBazaar and attempt to encrypt data.

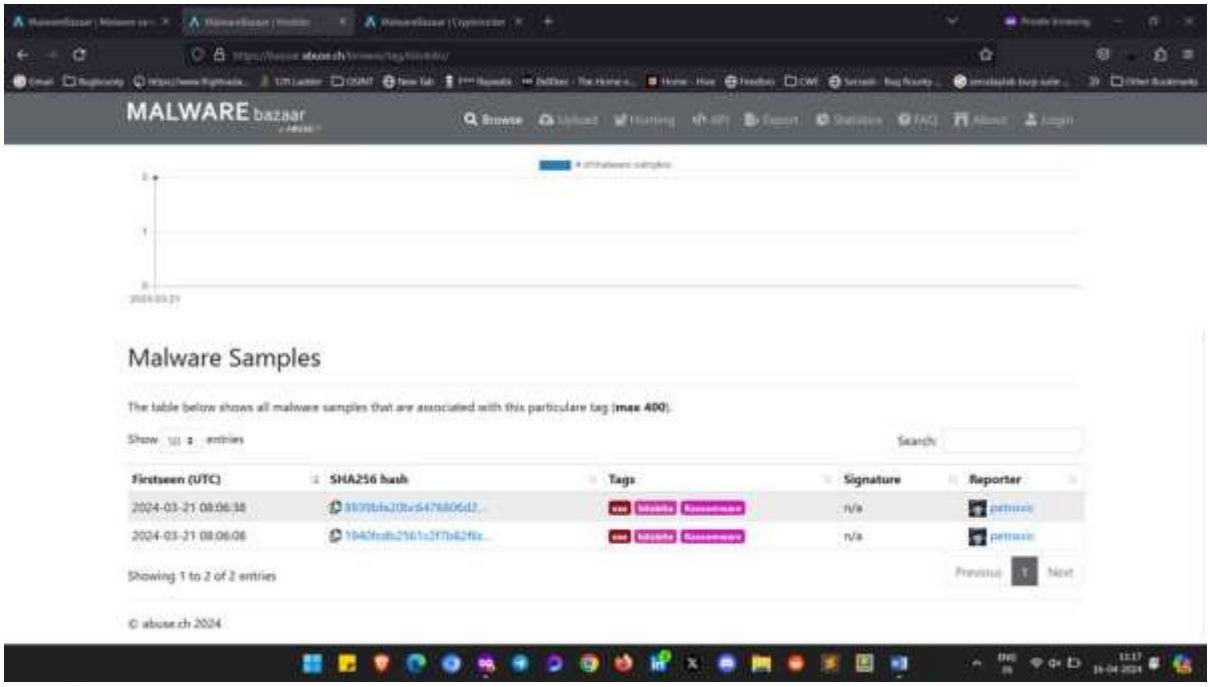

FIGURE 4. Proof of "hitobito malware"

Hitobito ransomware ran successfully.

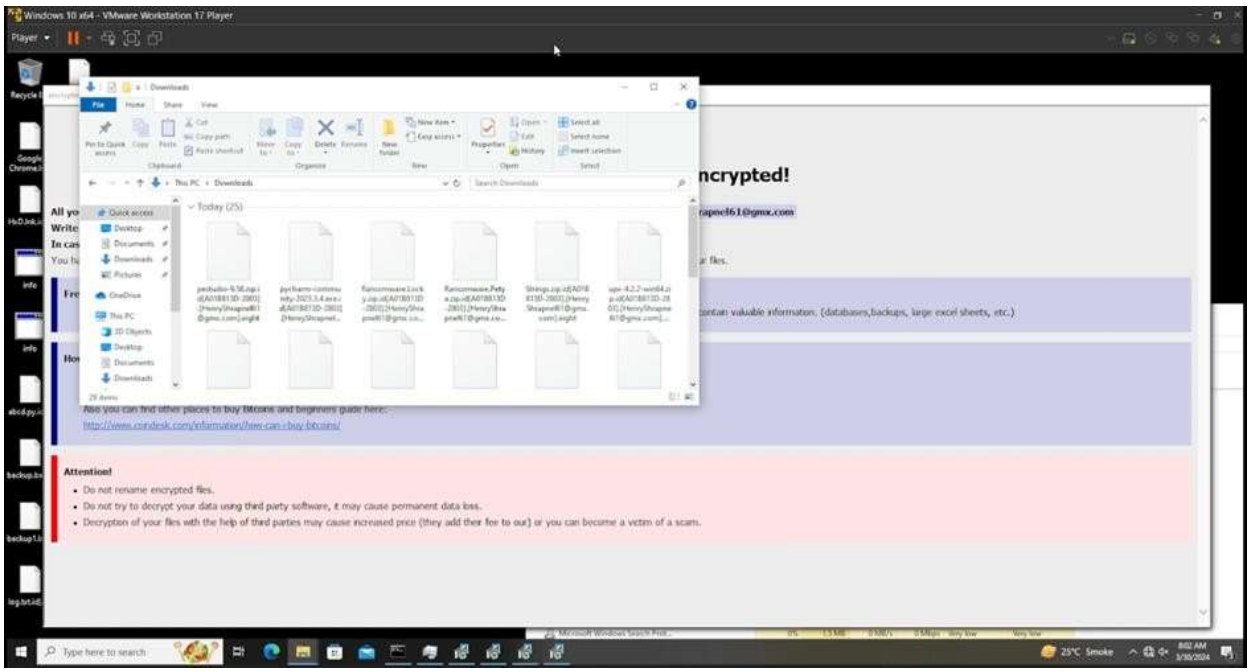

FIGURE 5. proof of "hitobito malware"

After running the Hitobito ransomware, the data becomes encrypted, but the backup file remains unaffected. The backup file is safe.

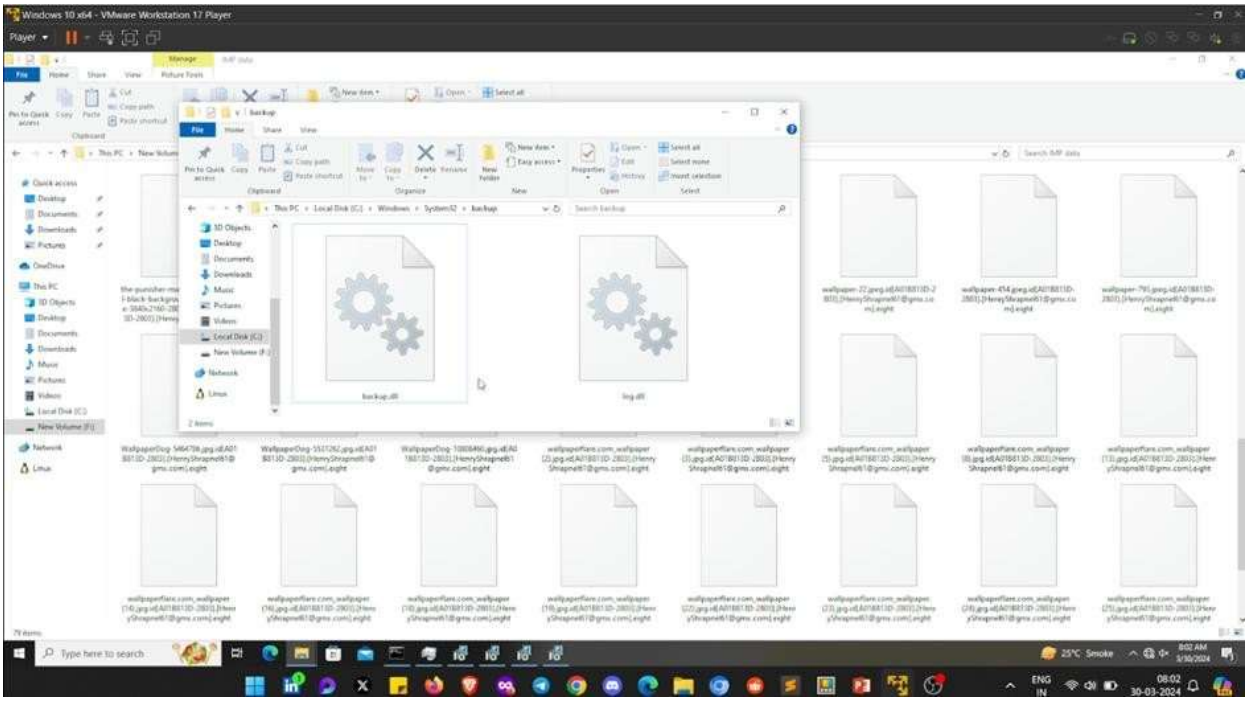

FIGURE 6. Proof of "hitobito malware"

Lockbit ransomware [first seen: 2023]

Download the LockBit ransomware sample from MalwareBazaar and execute it to test whether the data is encrypted or not.

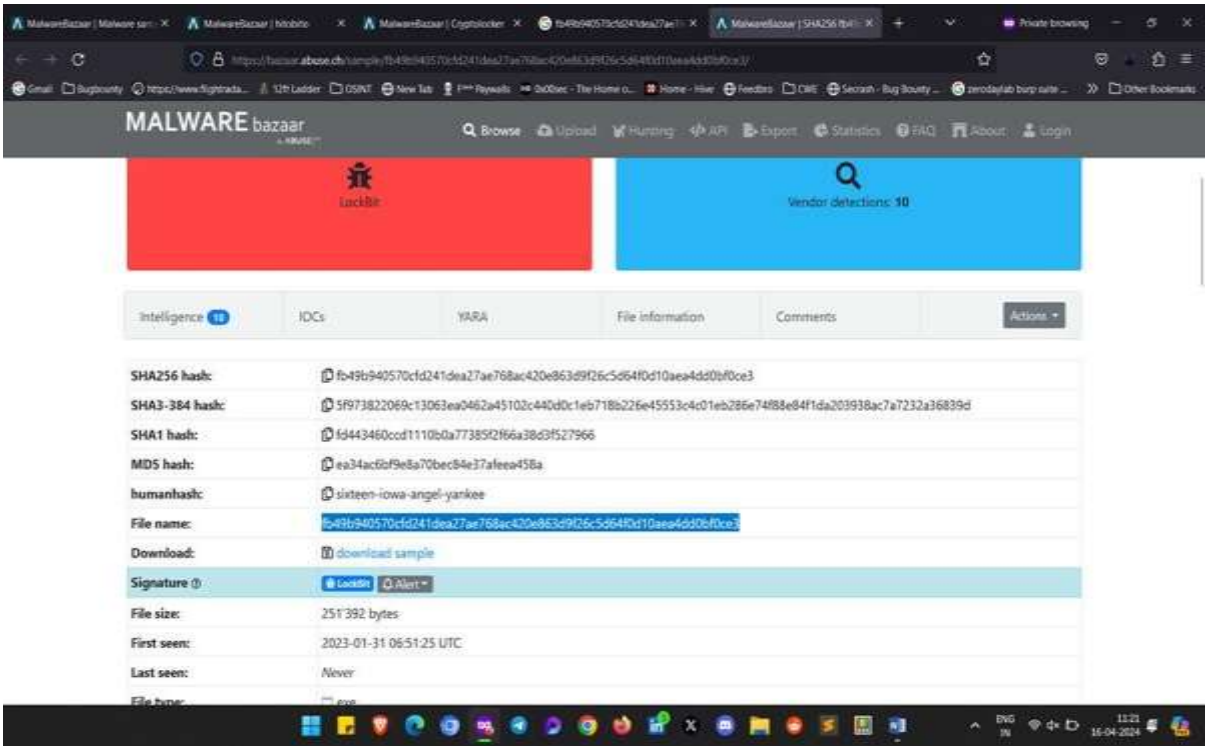

FIGURE 7. Proof of "lockbit malware"

After the execution of the LockBit ransomware, the data will be encrypted. However, our backup file remains safe in the system32/backup folder.

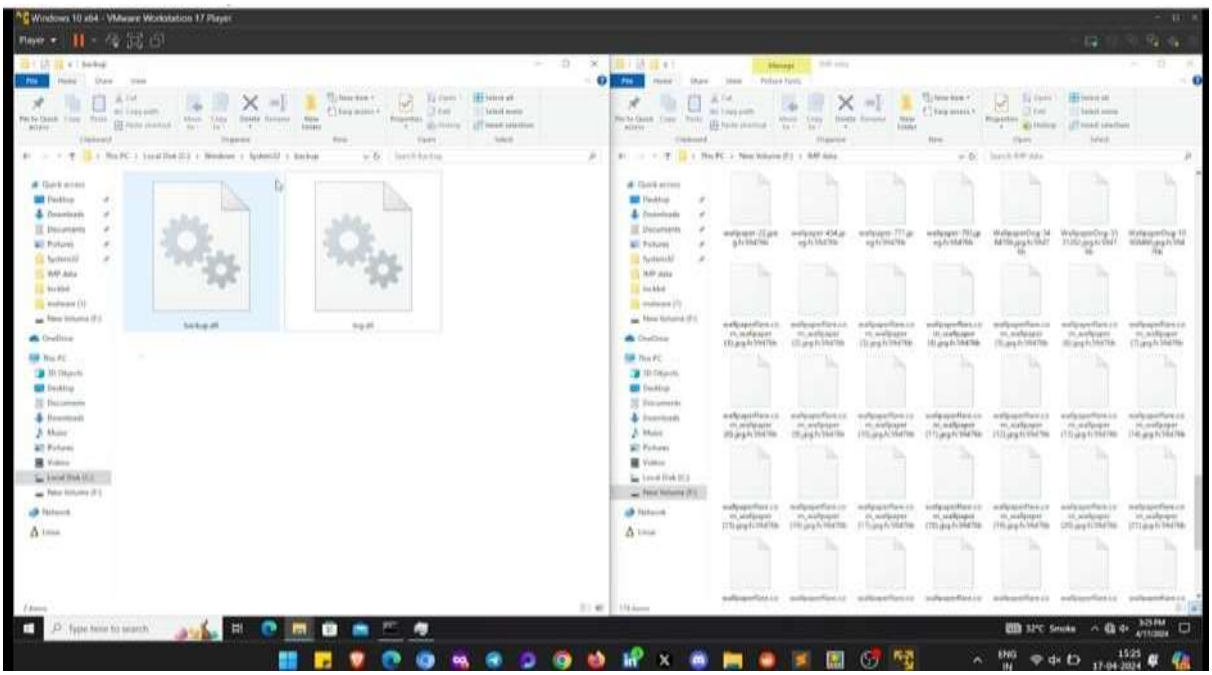

FIGURE 8. Proof of "lockbit malware"

MedusaLocker Ransomware [first seen: 2022]

Collect the MedusaLocker Ransomware sample from MalwareBazaar and execute it.

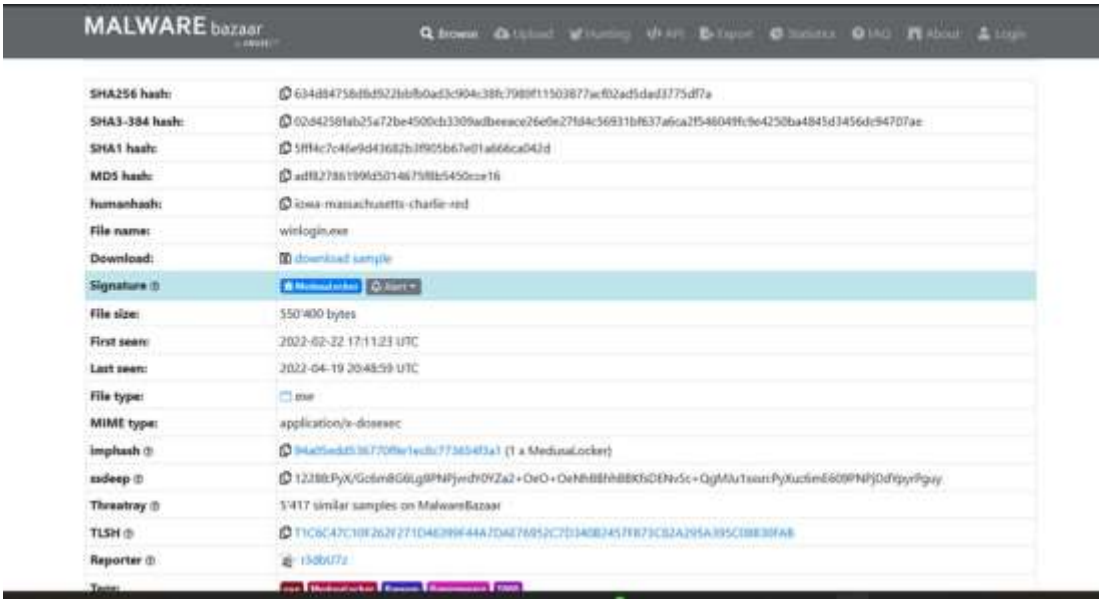

FIGURE 9. Proof of "medusalocker malware"

Medusa ransomware executes successfully.

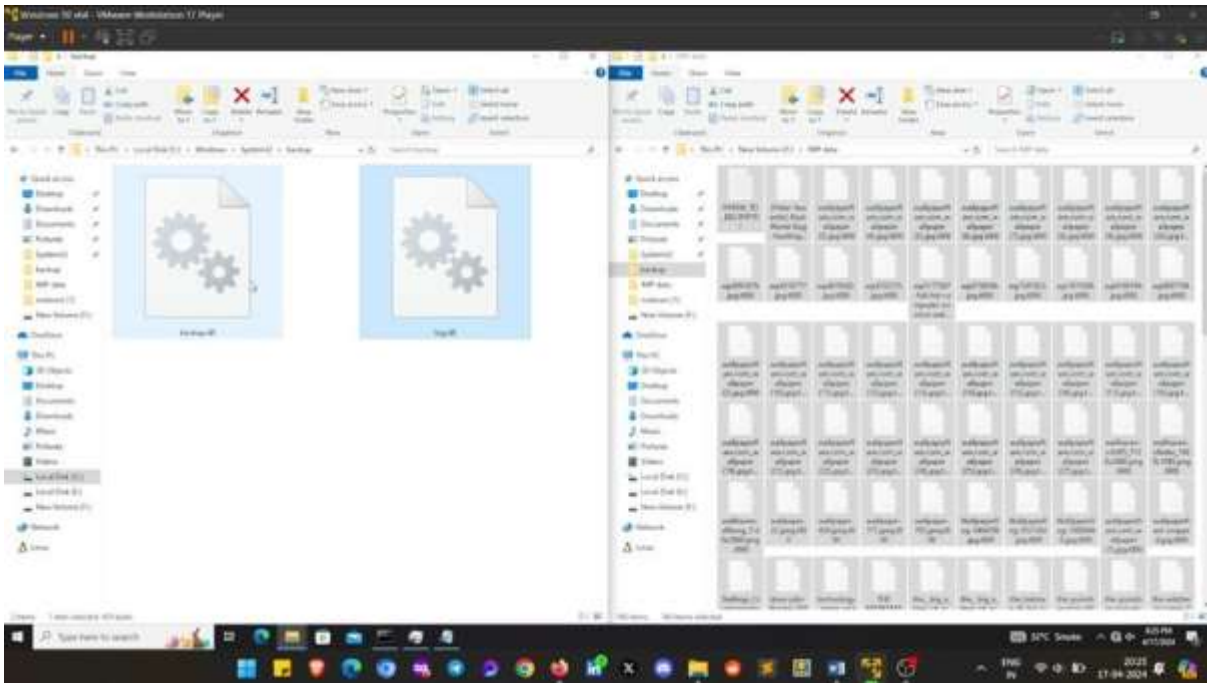

FIGURE 10. proof of “medualocker malware”

Our project successfully tested the latest ransomware over the past three years and executed it without compromising user data. We ensured the safety of important data by creating backups stored securely. These backups were created in the system32 and backup folders, with the source point being the user's important folder where their files were stored.

To enhance security, we implemented a hash password and a static password. The hash password adds an additional layer of protection, while the static password is required for users to access and recover their backups. This approach ensures that users can safely store and retrieve their important data in case of a ransomware attack.

V. DISCUSSION

Our project focused on protecting data from ransomware attacks, yielding several key insights from experimentation and testing. We observed that ransomware predominantly targets specific file types such as .txt, .jpg, and .pdf, leading us to recommend users store these files in a designated "imp work" folder.

A batch file, "backup.bat," automated the backup process from the "imp work" folder to a secure "backup works" directory in the system32 folder. Utilizing 7-Zip for compression and converting file extensions to .dll enhanced security. Log generation provided records of backup activities, including dates, times, and passwords.

A Python script, "abcd.py," offered static password protection by generating random passwords and converting them to hashes for backup sessions. Testing with various ransomware attacks confirmed the tool's effectiveness in safeguarding backed-up files and critical system components, ensuring the integrity of the "backup works" folder even when the system was compromised.

Additionally, the development of a recovery batch file, "recovery.bat," allowed users to retrieve backed-up files using the last generated password stored in "log.dll." While the tool is designed for Windows operating systems and has limitations regarding backup sources and supported file formats, it provides a reliable defence against ransomware, ensuring the continuity and security of user data.

VI. CONCLUSION AND FUTURE DIRECTIONS

This project developed a robust tool aimed at protecting user data from ransomware attacks. By advising users to store critical files in a designated folder named "imp work," our tool automates the backup process to a secure location within the system32 directory using "backup.bat." Utilizing 7-Zip compression and converting file extensions to .dll enhances security, supported by a Python script, "abcd.py," providing static password and hash protection. Testing against various ransomware attacks confirmed its effectiveness in safeguarding data. The "recovery.bat" file assists in

restoring files using the last password stored in a log file. While designed for Windows, the tool provides reliable defence, ensuring data safety and integrity.

Future enhancements for the tool include implementing advanced encryption algorithms to strengthen backup files and password protection. Expanding cross-platform compatibility to macOS and Linux will broaden user accessibility. Introducing automatic updates will ensure seamless delivery of bug fixes, security patches, and new features. Integration with cloud storage services such as Google Drive and Dropbox will provide offsite data backup options. Exploring machine learning algorithms for real-time ransomware detection and establishing a user feedback loop will enhance tool functionality and user experience. Developing educational resources to educate users about ransomware threats and collaborating with antivirus vendors for comprehensive protection are also planned. Additionally, creating a mobile application for managing backups on smartphones and tablets and forming partnerships with educational and cybersecurity organizations will promote tool adoption and usability.

REFERENCES

[1] S. Vasoya, K. Bhavsar and N. Patel, "A systematic literature review on Ransomware attacks," arXiv (Cornell University), 1 2022.

[2] J. Morris, D. Lin and M. Smith, "Fight virus like a Virus: A new defense method against File-Encrypting ransomware," arXiv (Cornell University), 1 2021.

[3] P. Joon-Young, "Data Protection Based on Hidden Space in Windows Against Ransomware," Proceedings of Sixth International Congress on Information and Communication Technology, pp. 629-637, 2022.

[4] M. M. Khan, M. F. Hyder, S. M. Khan, J. Arshad and M. M. Khan, "Ransomware prevention using moving target defense based approach," Concurrency and Computation Practice and Experience, vol. 35, no. 7, 1 2022.

[5] S. Lee, S. Lee, J. Park, K. Kim and K. Lee, "Hiding in the crowd: Ransomware protection by adopting camouflage and hiding strategy with the link file," IEEE Access, vol. 11, pp. 92693-92704, 1 2023.

[6] O. Aslan and R. Samet, "A comprehensive review on malware detection approaches," IEEE Access, vol. 8, pp. 6249-6271, 1 2020.

[7] S. Sheen, K. A. Asmitha and S. Venkatesan, "R-Sentry: Deception based ransomware detection using file access patterns," Computers & Electrical Engineering, vol. 103, p. 108346, 1 2022.

[8] H. Oz, A. Aris, A. Levi and A. S. Uluagac, "A survey on Ransomware: Evolution, taxonomy, and defense solutions," ACM Computing Surveys, vol. 54, no. 11s, pp. 1-37, 1 2022.

[9] R. Nagano, R. Hisasue, H. Inamura and S. Ishida, "Recovery Method for Ransomware Encryption Attacks with File Extension Changing on File Server," 2023 Fourteenth International Conference on Mobile Computing and Ubiquitous Network (ICMU), pp. 1-4, 1 2023.

[10] R. Moussaileb, N. Cuppens, J.-L. Lanet and H. L. Bouder, "A survey on Windows-based ransomware taxonomy and detection Mechanisms," ACM Computing Surveys, vol. 54, no. 6, pp. 1-36, 1 2021.

[11] P. H. Meland, Y. F. F. Bayoumy and G. Sindre, "The Ransomware-as-a-Service economy within the darknet," Computers & Security, vol. 92, p. 101762, 1 2020.

[12] A. Moser, C. Kruegel and E. Kirda, "Limits of Static Analysis for Malware Detection," Twenty-Third Annual Computer Security Applications Conference (ACSAC 2007), pp. 421-430, 1 2007.

[13] Q. Chen and R. A. Bridges, "Automated Behavioral Analysis of Malware: A case study of WannaCry ransomware," 2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA), pp. 454-460, 1 2017.

[14] E. Kirda, "UNVEIL: A large-scale, automated approach to detecting ransomware (keynote)," 2017 IEEE 24th International Conference on Software Analysis, Evolution and Reengineering (SANER), p. 1, 1 2017.

[15] A. Vehabovic, N. Ghani, E. Bou-Harb, J. Crichigno and A. Yayimli, "Ransomware Detection and Classification Strategies," 2022 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom), pp. 316-324, 1 2022.

[16] A. Ren, C. Liang, I. Hyug, S. Broh and N. Jhanjhi, "A Three-Level ransomware detection and Prevention mechanism," EAI Endorsed Transactions on Energy Web, vol. 0, no. 0, p. 162691, 1 2018.

[17] P. Sharma, S. Kapoor and R. Sharma, "Ransomware detection, prevention and protection in IoT devices using ML techniques based on dynamic analysis approach," International Journal of Systems Assurance Engineering and Management, vol. 14, no. 1, pp. 287-296, 1 2022.

[18] C. Constantinescu and S. Seshadri, "Sentinel: ransomware detection in file storage," ACM Computing Surveys, p. 1, 1 2021.

[19] F. Guvçi and A. Şenol, "An Improved Protection Approach for Protecting from Ransomware Attacks," Journal of Data Applications, vol. 0, no. 1, pp. 69-82, 1 2023.

[20] A. AlSabeh, H. Safa, E. Bou-Harb and J. Crichigno, "Exploiting Ransomware Paranoia For Execution Prevention," ICC 2020 - 2020 IEEE International Conference on Communications (ICC), 1 2020.

[21] N. Scaife, H. Carter, P. Traynor and K. R. B. Butler, "CryptoLock (and Drop It): Stopping Ransomware Attacks on User Data," 2016 IEEE 36th International Conference on Distributed Computing Systems (ICDCS), 1 2016.

[22] K. P. Subedi, D. R. Budhathoki, B. Chen and D. Dasgupta, "RDS3: Ransomware defense strategy by using stealthily spare space," 2017 IEEE Symposium Series on Computational Intelligence (SSCI), pp. 1-8, 1 2017.

[23] J. Ahn, D. Park, C.-G. Lee, D. Min, J. Lee, S. Park, Q. Chen and Y. Kim, "KEY-SSD: Access-Control Drive to Protect Files from Ransomware Attacks," arXiv (Cornell University), 1 2019.

[24] A. Pagan and K. Elleithy, "A Multi-Layered Defense approach to safeguard against ransomware," 2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC), pp. 942-947, 1 2021.

[25] J. Choi, J. Lee, G. Lee, J. Yu and A. Park, "A defense mechanism against attacks on files by hiding files," Journal of Korea Society of Industrial Information Systems, 2022.

[26] N. R. Maulana, N. B. B. S. Manurung, N. N. B. N. Putra and N. A. R. Kardian, "Efficiency Analysis of compression software (WINRAR and 7-ZIP) across diverse data types on Windows 11 and Ubuntu 23.10," Jurnal Info Sains Informatika dan Sains, vol. 13, no. 03, pp. 921-926, 1 2023.