*Computers, Materials & Continua*
DOI:10.32604/cmc.2021.015941
***Article***

Tech Science Press

# A User-friendly Model for Ransomware Analysis Using Sandboxing

**Akhtar Kamal[1], Morched Derbali[2], Sadeeq Jan[1,*], Javed Iqbal Bangash[3], Fazal Qudus Khan[2], Houssem Jerbi[4], Rabeh Abbassi[4] and Gulzar Ahmad[5]**

[1]Department of Computer Science & Information Technology, National Center for Cyber Security, University of Engineering & Technology, Peshawar, 25120, Pakistan
[2]Department of Information Technology, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah, 21589, Saudi Arabia
[3]Institute of Computer Sciences and Information Technology (ICS/IT), University of Agriculture, Peshawar, 25130, Pakistan
[4]Department of Industrial/Electrical Engineering, College of Engineering, University of Ha'il, Hail, 1234, Saudi Arabia
[5]Department of Electrical Engineering, University of Engineering & Technology, Peshawar, 25120, Pakistan
[*]Corresponding Author: Sadeeq Jan. Email: sadeeqjan@uetpeshawar.edu.pk
Received: 15 December 2020; Accepted: 17 January 2021

**Abstract:** Ransomware is a type of malicious software that blocks access to a computer by encrypting user's files until a ransom is paid to the attacker. There have been several reported high-profile ransomware attacks including WannaCry, Petya, and Bad Rabbit resulting in losses of over a billion dollars to various individuals and businesses in the world. The analysis of ransomware is often carried out via sandbox environments; however, the initial setup and configuration of such environments is a challenging task. Also, it is difficult for an ordinary computer user to correctly interpret the complex results presented in the reports generated by such environments and analysis tools. In this research work, we aim to develop a user-friendly model to understand the taxonomy and analysis of ransomware attacks. Also, we aim to present the results of analysis in the form of summarized reports that can easily be understood by an ordinary computer user. Our model is built on top of the well-known Cuckoo sandbox environment for identification of the ransomware as well as generation of the summarized reports. In addition, for evaluating the usability and accessibility of our proposed model, we conduct a comprehensive user survey consisting of participants from various fields, e.g., professional developers from software houses, people from academia (professors, students). Our evaluation results demonstrate a positive feedback of approximately 92% on the usability of our proposed model.

**Keywords:** Ransomware; sandbox; user-friendly model; survey

## 1 Introduction

With the increase in the use of internet via mobile phones and computer systems, cyber-crimes have also been increased. In particular, the use of Information and Communication Technology

(ICT) for financial purposes have also increased the risk of cyber-attacks on such infrastructures. According to the reports of Federal Bureau of Investigation (FBI), several ransomware attacks were used for collecting $209 million in the first 3 months of the year 2016.

Ransomware is one of the most pervasive and destructive threats to individuals and organizations. In this attack, the attackers can take control of the target computer and encrypt the stored files and applications [1]. Once the encryption process is successful, the attackers then demand the victims for the payment of ransom to restore (decrypt) the encrypted files. The payment is usually carried out via Bitcoins which is a secure and untraceable method for money transfer. After the payment of ransom, the control of the PC is returned to the victim. During a ransomware attack, the victim should first attempt to stop the ransomware from further potential damage to the other connected systems, mailboxes, shared files, and drives. This can be performed by disconnecting the victim system from the internet immediately after the attack is detected. Next, the lost/encrypted data should be restored from the backup (if available). However, this is often not possible especially if the ransomware has already encrypted the whole computer system of the victim. Also, most people do not keep regular backups of their data. Therefore, it is important to understand the taxonomy of ransomware and the existing approaches/tools for analysis of such files.

Sandbox environments are widely used to analyze ransomware files, however, the initial installation setup and understanding of the generated reports are often difficult tasks. Furthermore, the key information required for the detection of ransomware is difficult to locate in the generated reports of such environments. This research work aims to provide a user-friendly model to ease the processes of ransomware file submission for analysis and producing the summarized reports that are easy to understand for ordinary computer users. The generated report will contain the key information that can be used to understand the taxonomy of ransomware. For the implementation, we use Cuckoo which is one of the well-known malware analysis systems for ransomware. For the front-end of our tool, we use HTML and CSS, while the REST APIs are used for file submission to Cuckoo server and fetching of the summarized report.

The remainder of this paper is organized as follows: Section 2 contains the background on ransomware. Section 3 presents the related work on ransomware detection methods. The research methodology is explained in Section 4 while Section 5 discusses the results. Finally, the conclusion and future work is presented in Section 6.

## 2 Background

Ransomware is a type of malware that aims to perform the encryption process on a remote or target computer and blocks user access to the data until a ransom is paid to the attacker [2]. Encryption process is performed by ransomware on various types of data residing in the target system. For exploiting the target machine or system, attackers often use social engineering techniques. The delivery or infection can be done through multiple attack vectors, such as exploit kits, malicious pdf files or MS Office files, phishing, and malicious advertisement. Fig. 1 illustrates a typical ransomware attack setup. In most cases, ransomware gets inside the system when the client or user clicks on the links sent in the phishing emails. Once the malicious link is clicked, the payload is downloaded automatically in the backend and the execution process starts. To hide its identity, the ransomware does not execute as a standalone process but rather it uses a host file called dropper file. For example, it may use the Windows explorer process at the front while a genuine-looking process (e.g., “svchost”) in the background. This also helps the ransomware to

keep running on the infected systems, persist across reboots, and execute even if the system is started in "safe mode."

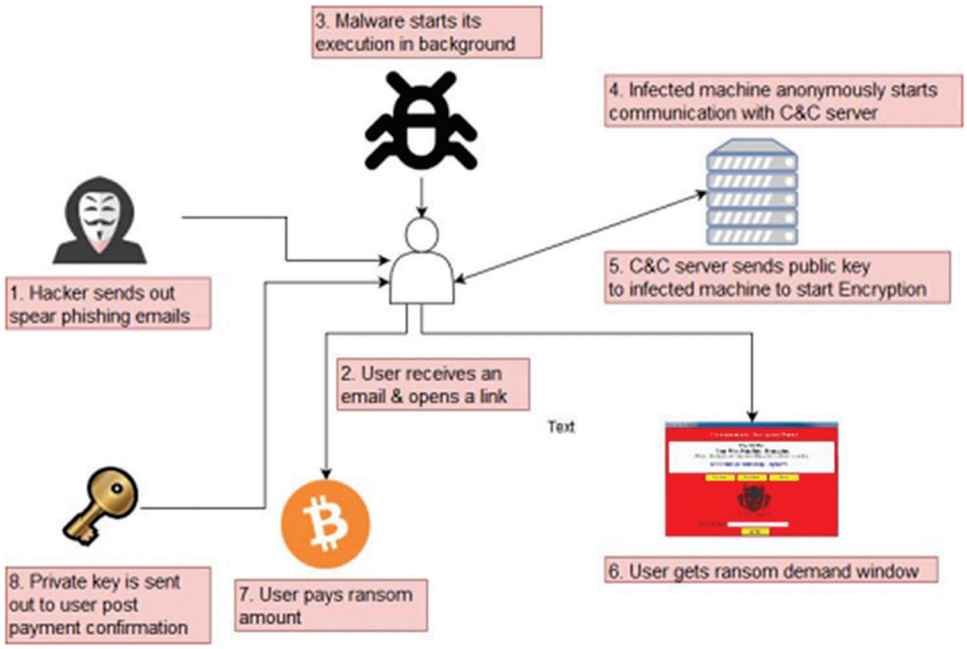


**Figure 1:** Ransomware execution cycle

To become persistent across reboots, the ransomware creates registry key additions (in windows) and also adds itself to the group of system startup processes. After installing itself on the victim's machine, the ransomware payload communicates with the Command and Control (C&C) server, which is operated remotely by the attacker. The C&C server confirms the incoming request from the infected system and generates a set of keys, consisting of a public key and a private key. The public key is sent to the ransomware payload which performs the encryption process on the target machine. The files encrypted with the public key can only be decrypted by using the private key which is held on the C&C server. The communications between the C&C server and the infected machines are protected by The Onion Router (TOR) browser. The ransom note notifies the user that his/her machine has been infected and can only be recovered by paying the ransom money. Payment directions are also provided, and normally, in these cases, a new and unique virtual currency address is originated for each user to make transactions undetectable. Following are some of the well-known types of ransomware and their history.

Bad Rabbit ransomware affected many organizations in Russia and Europe. It was spread by a flash player update on the compromised websites [3]. When a target machine was affected by Bad Rabbit, the user would be directed to pay 300$ in ransom. In these cases, the target was visiting a legitimate website, and a malware dropper file was then downloaded from the threat host. No exploits were used in this method, so the victim would have to manually execute the malware dropper, which pretends to be an Adobe Flash application. The analysis confirmed that Bad Rabbit uses the Eternal Romance exploit as an infection vector to spread within corporate networks.

Cerber was launched to target the online users of Microsoft 360 Office [4]. Millions of people were affected, demonstrating the attacker's use of a large-scale phishing campaign. The

key feature of this ransomware is the offline working and no need for internet connection for encryption. The ransomware uses RC4 and RSA Algorithm for file encryption. Furthermore, it contains VB scripts for exploitation. A decryption application for the Cerber is now available however its functionalities are limited as some of the information tends to be overlooked during the decryption process [5].

Crysis is another famous type of ransomware using the RSA and AES-128 algorithms for encryption [6]. Decryption tools are available for the older version; however, the new version of this ransomware is still not decryptable. This kind of malware encrypts data of fixed, removable, and network shared partition, by using a strong encryption algorithm, and makes it difficult to decrypt in a reasonable time. It was spread as an attachment to a legitimate file via emails.

Jigsaw is a type of ransomware that auto deletes the user's file after a specific interval of time [7]. It encrypts all files and then deletes the first file after an hour. Similarly, it deletes more files in the next hour and so on until the ransom is paid. Within 72 hours, all files are deleted from the target system. Jigsaw encrypts files with the normal extensions e.g., JPG, JPEG, GIF, PNG, BMP, ASP, SQL, CPP, CS, PHP, JAVA, DOCX. However, the decryption technique for Jigsaw is freely available.

Similar to other malwares, Locky was also spread through email in an invoice format using Microsoft word file [8]. After receiving the email and opening the invoice file, a pop-up file gets executed stating the requirement to enable the macro. Once the user enables the macro option, the Locky ransomware starts working and encrypts all data with the AES encryption algorithm. After the encryption is completed, and the user tries to open the encrypted file, it will direct to a website to pay for decryption. The encrypted file will be represented with the extension of Locky, zzz, and Asian, etc.

Spider ransomware contains malicious macros within a file and spreads them via email [9]. When a user downloads and opens the file, the malicious macros is executed and initiates the spider to start the encryption process. The malicious Microsoft word file contains the obfuscated source code, which enables the power shell of the target user to download and run payload at the target host. Next, the power shell script decodes the source code and executes it. This adds a spider extension to all encrypted files and start deleting files automatically after 96 hours.

GoldenEye ransomware is similar to Petya ransomware and it has caused financial damages to many organizations in Europe and USA [10]. It also works by using a Microsoft Office file containing malicious macros. These macros contain encrypted malicious file which runs when the macros are enabled. Goldeneye ransomware encrypts the file and adds an 8-character extension to each file.

Lime ransomware attack surfaced in 2018 offering manual standalone exploitation as well as embedding it within any legitimate software [11]. The lime ransomware had a unique function in comparison to the other ransomware, i.e., even after the payment of ransom and decryption of user files, it used to create a back door into the target system to again encrypt the files in future. Furthermore, Lime doesn't share any key with their exploited user instead they handle the internet-connected client remotely.

Teslacrypt is a type of ransomware that was used to destroy systems in USA, Germany, France, Italy, Spain and United Kingdom [12]. Similar to the other ransomware, it uses the AES encryption algorithm. Furthermore, it uses the Angular Exploit kit which specifically exploits the Adobe vulnerability. Once the vulnerability is detected, Teslacrypt starts execution. Teslacrypt

installs its files in the Microsoft Temp Folder and encrypts all types of data, e.g., JPG, DOXS, PDF, executable.

In 2017, Wannacry ransomware was discovered affecting a wide number of organizations all over the world [13]. Approximately 1,250,000 organizations were affected in over 150 countries. WannaCry has also been known for affecting Windows machines using a Microsoft Exploit kit. It also uses the EthernalBlue exploit of Windows Server Message Block (SMB). When WannaCry runs on a machine, it first encrypts all data and then scans the connected PC's in the local network and attacks them using the SMB vulnerability. It has also been termed as a network worm due to this feature.

## 3 Related Work

Various ransomware detection methods have been proposed in the literature [2–5]. In particular, Jethva [2] evaluated the static detection method of ransomware by modifying packed portable executables. To overcome the drawbacks of such classic signature-based detection systems, researchers have published several proposals on dynamic ransomware detection methods. For instance, machine learning with static analysis was used in [6] to detect various exploits. The authors treated Portable Executable (PE), strings information, and byte sequences of binary to categorize the various exploits using the Naïve Bayes classification algorithm. In their effort, they proposed a similar approach to classifying various binaries using n-gram byte sequence with different classification algorithms, which include naïve Bayes, decision trees, SVM, and boosting.

In a study conducted by Kara et al. [12] on ransomware detection, the average detection rate for new ransomware using static analysis was significantly low; only ten engines out of sixty tested could detect ransomware. Moreover, static detection systems can be evaded using the code obfuscation method. The authors explored this limitation of static ransomware detection and observed that advanced static-based detection could easily be evaded.

Crypto Drop is an early warning detection system to alert users during suspicious file activities [13]. The system mainly focused on monitoring user data for changes. The authors divided ransomware into three major classes: A, B, and C based on their encryption process. They treated similarity functions to measure the dissimilarity between the original and the encrypted contents of each file. Crypto Drop was unable to determine the purpose of the changes in its audit. For example, it was not able to differentiate between user-triggered encryption and ransomware triggered-encryption. The experimental evaluation was based on a dataset involving 582 ransomwares from 11 different families. An accuracy of 96.3% was obtained using dynamic analysis with a limited number of features. Additionally, most of the features treated in this system were binary. The authors focused only on the absence or presence of some of the features like registry key operations, Dynamic Link Library (DLL) operations, mutex, etc. However, in the new variants of various ransomware, the absence of these particular operations makes the detection model ineffective. For example, a registry key operation treated in one variant of ransomware might not be treated by the other variants or new versions of ransomware.

Chen proposed an approach for ransomware detection based on dynamic API calls flow graph by monitoring API call sequences of various binaries and converting them to a set of features [13]. They tested various data mining algorithms including random forest, SVM, Naive byes, and logistic regression. The logistic regression achieved the highest accuracy of 98.2% with the lowest false positive rate of 1.2%. However, the focus was only on a single feature to detect ransomware and the evaluation was based on a dataset consisting of only 168 ransomware

samples. In a follow-up effort presented in [14], a ransomware detection system called UNVEIL was proposed. UNVEIL looks at the filesystem layer to spot the typical ransomware behavior. It uses a text analysis method to detect ransomware threatening notes and continuously takes screenshots of the desktop to keep a check on potential screen lockers. It also uses statistical analysis based on the usage of memory, processor, and disk I/O rates to detect abnormal behavior for ransomware variants. The experimental evaluation resulted in 96.3% accuracy in detecting ransomware. Despite achieving relatively high accuracy, the model does not have early detection capability for ransomware attacks, nor does it provide any backup mechanism. Also, the proposed system is inherently reactive and ineffective for newer ransomware samples.

The boosted decision tree algorithm reached the best performance with a True-Positive Rate (TPR) of 98% and a False Positive Rate (FPR) of 5% [15]. Program opcodes were extracted from various binaries and arranged into a sequence. The published study contains some interesting signature designs of various ransomware that helped increase the false positive rate and a false negative rate of the classifier. The authors treated information gain (IG) to select valuable features and applied the SVM algorithm for classification. Experimental evaluation yielded a true positive rate of 81.40% and a false-positive rate of 2.67%. However, mostly, the classification systems relying only on static detection cannot detect new variants of ransomware.

Researchers have studied machine learning approaches with dynamic analysis for ransomware attacks occurred in the period of 2006–2014 [16–18]. In [16], authors explored 15 different ransomware families and observed that almost 94% of ransomware samples implement simple locking or encryption method. The authors suggested that by closely monitoring file system activity and the types of I/O request packets to the file system, it is possible to detect ransomware attacks. They also observed that Bitcoin addresses used to collect ransom payments from victims share similar transaction records, e.g., a small number of transactions, small Bitcoin amounts, short activity periods, etc. Despite proposing possible strategies for ransomware detection, no concrete experimental evaluation has been carried out [19]. An asset classification technique to assess the security of a web-based system is also presented in [20] that can be used to prepare a system for protection against ransomware attacks.

All of the above-mentioned existing ransomware detection techniques are not much effective and do not produce precise reports for analysis. Detecting ransomware via signature detection and machine learning techniques require several samples of the malware similar to the ransomware infecting your device. Additionally, other methods such as hashing and entropy techniques are complex, expensive and, in some cases, can result in a huge delay before the encryption operations are detected on a device. Finally, honeypot techniques rely on the tripwire files which can be coded, and has also a large delay before detecting encryption operations. These limitations in existing techniques make them ineffective against the timely detection of ransomware attacks. In contrast, our proposed approach is user-friendly, effective and can consistently detect ransomware at the initial stages of its lifecycle.

## 4 Methodology

Our proposed methodology of analyzing ransomware using Cuckoo Sandbox is presented in Fig. 2. It consists of three steps: (1) Problem Investigation, (2) Design and Development, and (3) Testing and Performance Evaluation.

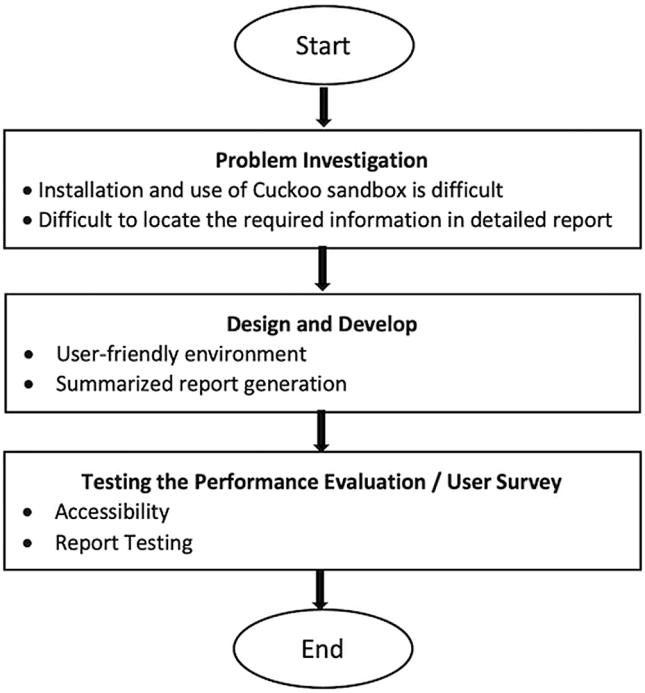


**Figure 2:** Proposed research framework

### *4.1 Problem Investigation*

In this step, we investigate the problem in the existing system by studying related research papers. We distinguish the performance of different techniques that tackle the malicious applications using the honeypot technique by defining rules and boundaries in the system. An efficient way to understand the malicious files is the preprocessing of the suspicious file in an isolated environment where the system is protected from the risks of malicious files. The cuckoo sandbox provides an isolated environment that can analyze different formats of files, e.g., PDF, EXE, DOC to identify malicious and different exploit kits. Cuckoo sandbox is Linux-based and is difficult for an ordinary computer literate to perform its initial installation and configurations.

### *4.2 Design and Development of User-friendly Environment*

In this step, we design the architecture of the proposed user-friendly model that can utilize the cuckoo sandbox. An HTML form is created which contains two buttons, i.e., one for the file attachment and another to submit the file to the cuckoo server. This form allows remote users to easily understand the method of file submission. Next, the file is uploaded to the cuckoo submission folder using the REST API. The server continuously performs the analyses on the submitted files present in the folder. Moreover, each file in this folder is assigned a unique ID. After analyzing the file, a detailed report is generated in the result folder and the summarized result is obtained from the server via REST API.

### *4.3 Testing and Performance Evaluation and Usability Survey*

Following the Design and Development step, we will execute the developed tool and check the remote accessibility options to ensure that it accepts and analyzes the file. In addition, we will also test the static and dynamic analysis of the cuckoo sandbox [21]. Moreover, we will test to ensure

that the generated summarized report for the client contains major functions and activities of the malicious file. In the last stage, there will be a usability survey about system accessibility and performance. This survey will include participants from software houses, security labs including the National Center for Cyber Security (NCCS-UETP) University of Engineering & Technology Peshawar, Pakistan, and other University Students. The survey questionnaire is conducted using Google Forms. This is due to the fact that the form is easy to send to the target users and general university users and the form automatically calculates the results of the survey as well.

## 5 System Model

The ultimate goal of the proposed system is to provide a user-friendly environment to analyze malicious files and also provide a summarized and precise report. The proposed system model is depicted in Fig. 3. As can be seen in the Figure, users from any platform (Windows, Android, Linux, Iphone, etc) can submit the ransomware files to the front-end system which offers a user-friendly interface. The front-end system then uploads the files to the Cuckoo server via Internet. The results are sent back to the front-end system which summarizes and presents the required information to users. The front-end system performs all the processing on the input files and the received generated reports from the Cuckoo server.

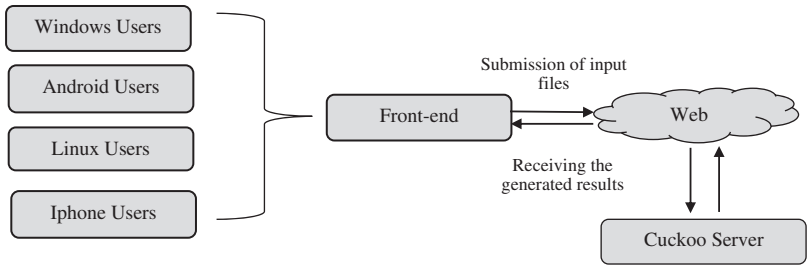


**Figure 3:** Proposed model

## 6 Results

The evaluation results of our proposed model are explained below.

### *6.1 Accessibility*

For ease of access, our HTML-based form contains only two buttons, i.e., one for attaching the file and a second button for submission. Next, the selected file is uploaded to the cuckoo server via REST API. The accessibility of the cuckoo server does not need any special operating system, i.e., users of all operating systems like Android, Windows or Linux can easily submit a file to the cuckoo server. Users can submit any file for analysis by using the HTML form. The form is depicted in Fig. 4.

### *6.2 Report Generation*

The uploaded file is saved in the cuckoo task folder where a unique ID is assigned to each file that is further analyzed keeping the same ID. Next, a detailed report is generated, however, it is difficult to understand for an ordinary user. A summarized report generated by our work is depicted in Fig. 5 which shows all the major functions and activities of ransomware detected at run time. This figure is a summary report which shows only the specific detail detected by the

cuckoo server. The last line "drops 99 unknown file mime-types which indicate of ransomware and writing encryption files back to the disk", clearly shows that the encryption process is started and data is sent back to the disk thus identifying the function of the ransomware. Furthermore, the report also shows that it installs itself in the Windows startup folder for autorun. The main reason for installing itself in the windows startup folder is to make itself persistent. Once the system is infected whenever it restarts, the malicious file will automatically run from the startup folder.

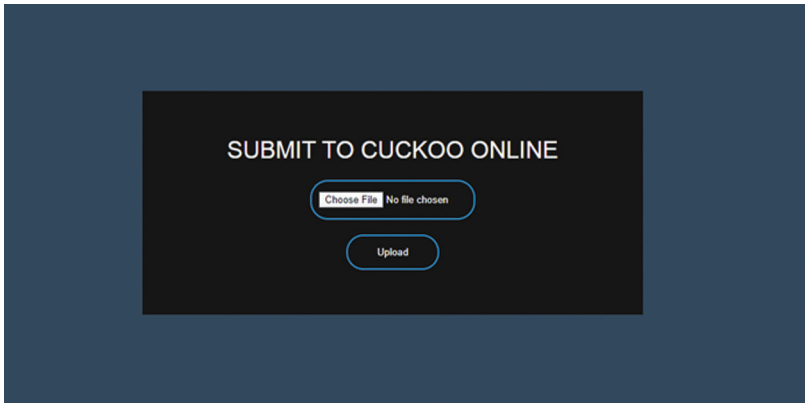


**Figure 4:** Front-end of Cuckoo user-friendly model

Moreover, the report shows that it creates a thread using remote thread in a non-child process, this is indicative of process injection. As depicted in Fig. 5, the remote thread creation is detected showing that the remote machine or attacker is using the hacked machine. The next line shows that allocate executes permission to another process indicative of possible code injection. The process injection technique is used to bypass antivirus, from the detection of malicious files, the antivirus scanning running process can not recognize the malicious file because malicious file injects itself in another legitimate file during in memory. Moreover, the messages are related to the malicious file as mention in the report "a process attempts to delay the analysis task", drop an executable file to the App folder, hidden process, searching the running process potentially to identify a process for sandbox evasion code injection or memory dumping.

### *6.3 Analysis of Survey Results*

Our survey consists of 17 questions divided into two portions. The first portion contains general questions regarding the information about user knowledge of the virus files and tools used for the analysis of the malicious files. The second portion of the survey is related to our developed model, i.e., "user-friendly model to ease the detection and investigation of ransomware using cuckoo sandbox" to ensure the accessibility, usability and results generation of the model. Furthermore, participants of our user survey belong to various software houses, employees of the cyber security research labs particularly the National Center for Cyber Security (NCCS)-UETP, University of Engineering & Technology Peshawar, Pakistan and other University Students.

The responses for questions 1–9 are shown in Fig. 6 while Fig. 7 shows the responses of the remaining questions, i.e., 10–17. Regarding the response of question 1 about the knowledge of the virus, as shown in the figure, 37.2% of people don't know about the actual function of

a virus while 52.7% of people have some knowledge about it. Only 20% of users possess good information on viruses.

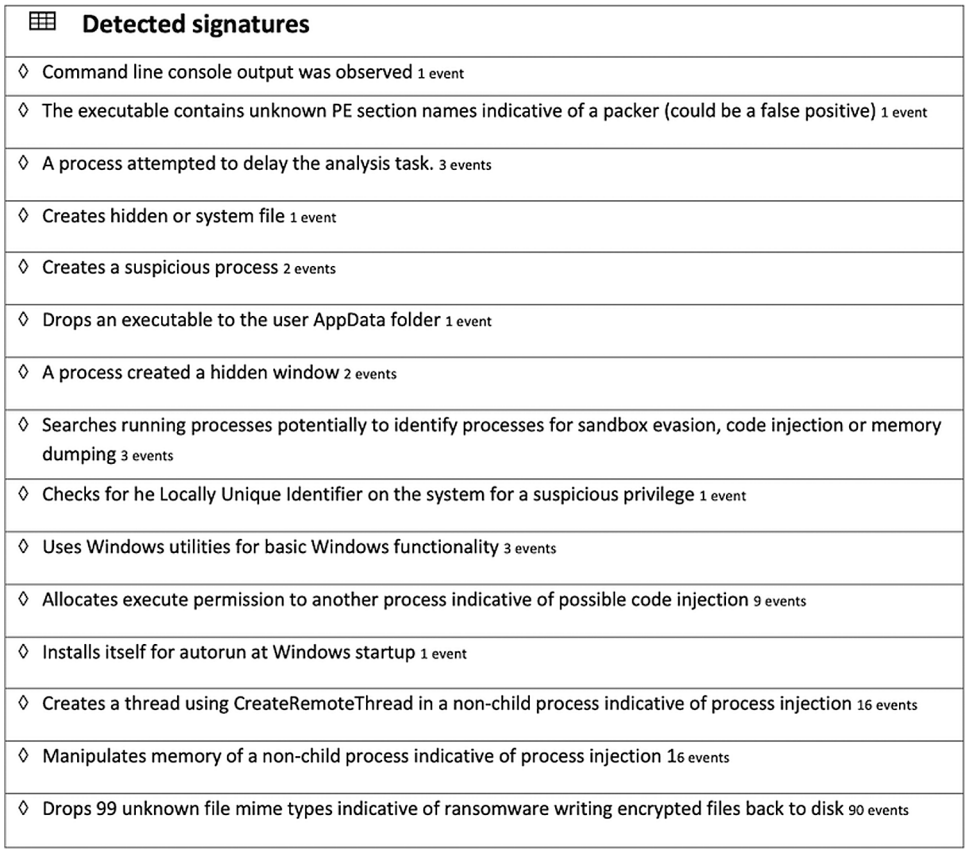


**Figure 5:** Generated summarized report

In question 2, we identify the number of people who know the protection steps against viruses. Surprisingly, around 39% of people have no idea of how to protect themselves from viruses. Similarly, 40.7% of people are able to follow a few steps to protect themselves from such malicious software attacks. On the other hand, only 6% of people are more experienced and have all the required knowledge for protection.

In 3rd question, we want to investigate the awareness of malware analysis in general. The result of this question was also interesting, i.e., around 81.3% of people responded that they do not know anything about malware analysis. On the other hand, only 18.7% of people are confident that they possess good knowledge of malware analysis.

Question 4 is related to the experience with malware analysis where the result shows that that around 92% of people have no experience in any kind of malware analysis. Only the remaining 8% people responded with a positive answer regarding their knowledge about malware analysis.

Question 5 is similar to the previous question, however, here we ask specifically about the user experience of using any kind of tool for malware analysis. The result shows that 83.3% of people have no idea about any kind of tools that are specifically used for malware analysis, while the remaining16.7 people have at least some knowledge of tools that can be used for malware analysis.

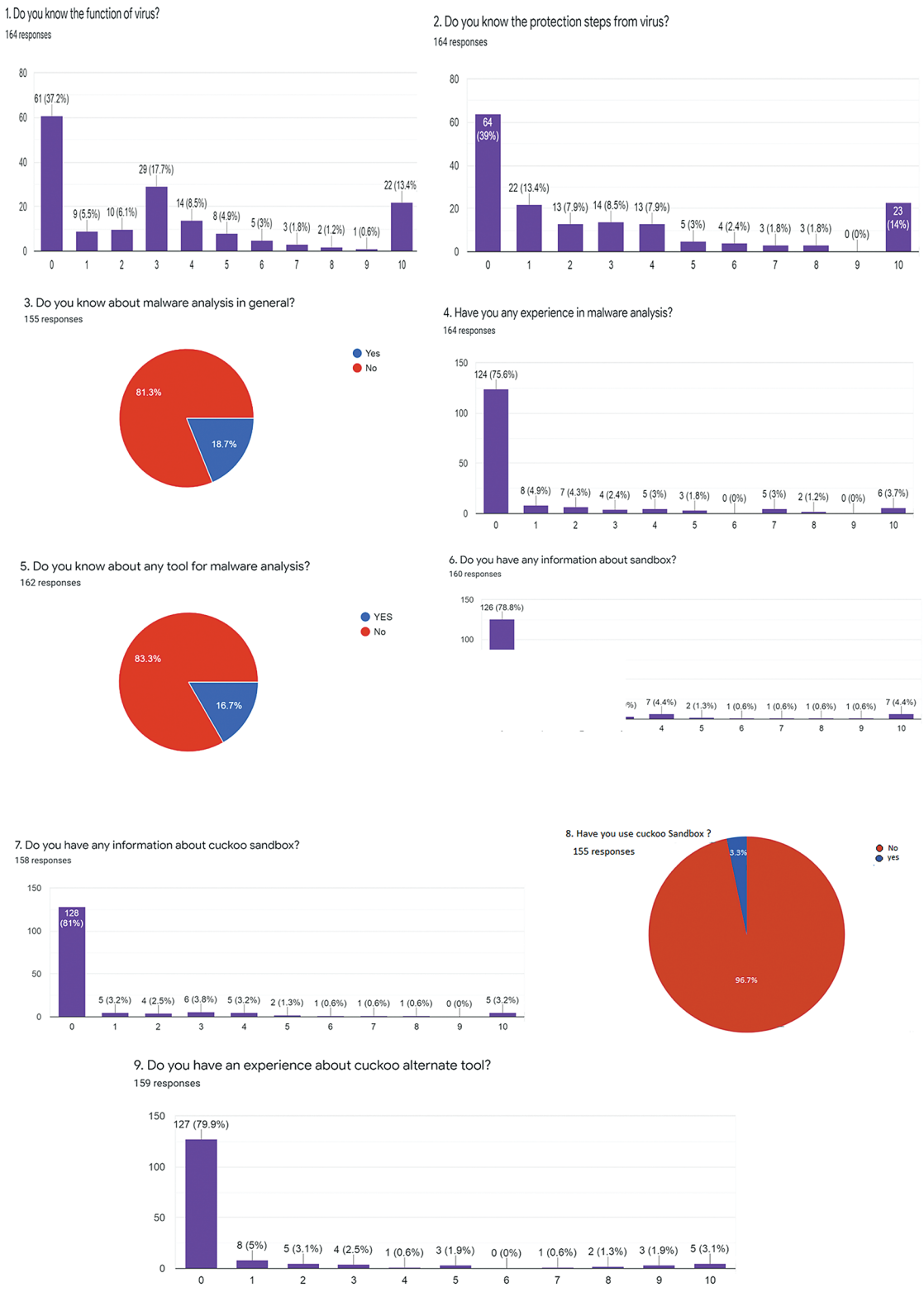


**Figure 6:** Knowledge about malicious files

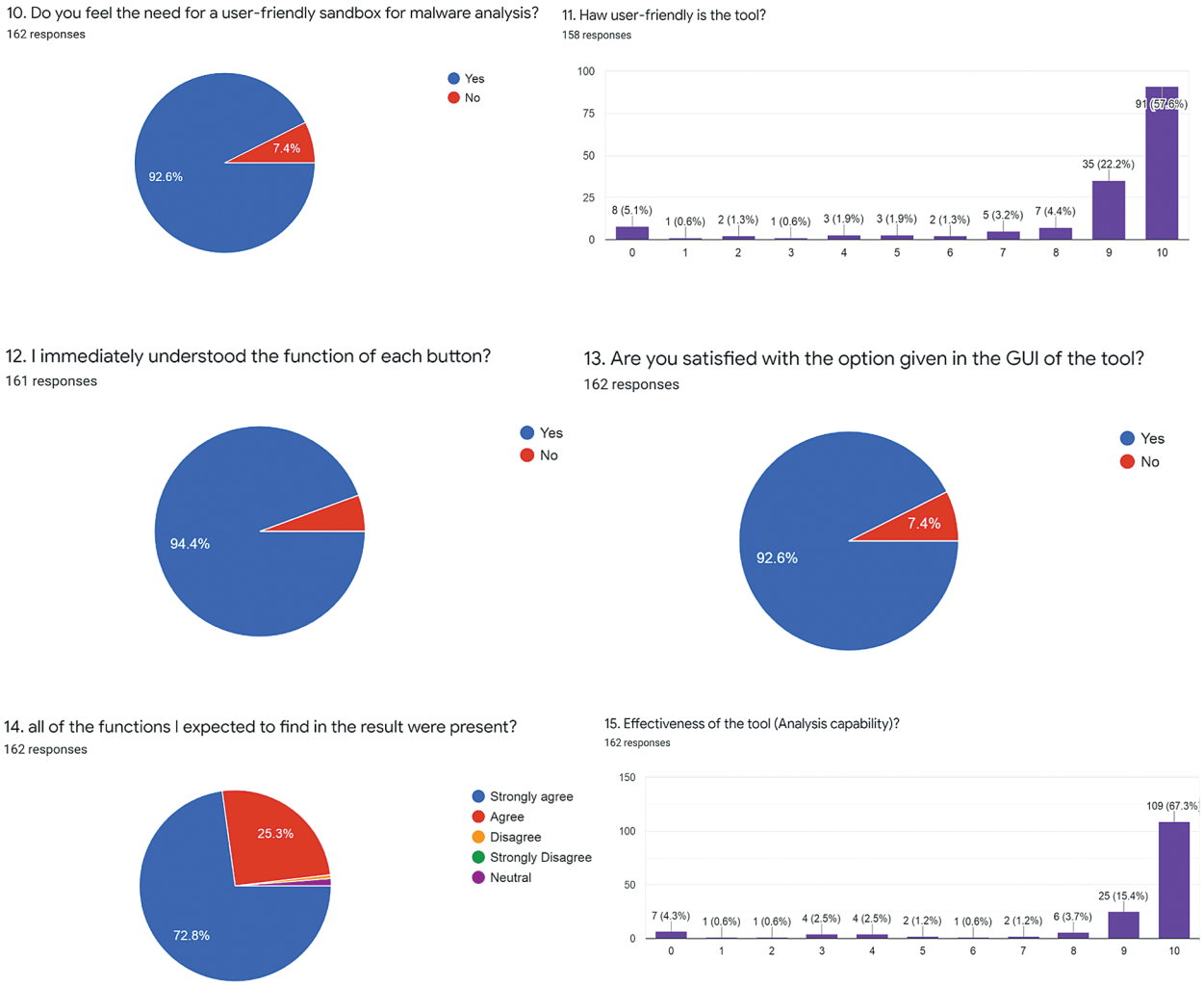


**Figure 7:** Performance and usability

Questions 6–8 are about sandbox environments and the Cuckoo sandbox. The result of questions 6 and 7 showed that a very large number of people don't know about sandbox and in particular, have no knowledge about the Cuckoo. Unfortunately, only 6.4% of people know the use of sandbox and 4.2% have knowledge about the use of cuckoo sandbox. In contrast, 78.8% and 81% of people do not know the function of sandbox and cuckoo sandbox respectively. Similarly, the response of question 8 shows that only 3.3% of users can use the cuckoo sandbox while the remaining (96.7%) are not familiar with the use of the Cuckoo sandbox in any context.

The last question (9) of this part is about the knowledge of any alternate tools for Cuckoo sandbox. Interestingly, only 3.1% of the user has experience of alternate tools for Cuckoo to analyze the malicious files while the remaining 96.9% of the users have no experience with any alternate tools but only the cuckoo sandbox.

## 7 Conclusion and Future Work

In this research work, we have developed a user-friendly model to analyze ransomware using a sandbox environment. For evaluating our proposed model, we have used the Cuckoo sandbox environment which is a leading open-source automated malware analysis system. The front-end of the model has mainly two components, i.e., the file selection module and the subsequent submission module. The created front-end is based on HTML and CSS, while on the backend of the HTML form, a REST API is used to submit the file to the Cuckoo task folder. Furthermore, the REST API is used to fetch only the important results from the detailed report. A carefully engineered virtual machine environment is used to provide an isolated space for analyzing the ransomware.

We further evaluated our proposed model by conducting a comprehensive user survey to ensure the usability of our developed user-friendly modules and the evaluation of generated results by ordinary users. First, we collected answers to some general questions including the function of viruses, the required protection steps, analysis of malicious files, the function of sandboxes, the use of cuckoo sandbox, and any alternate relevant tools. In the next part of our survey, we asked questions about the ease of use of our developed GUI, effectiveness, efficiency, and the summarized report generation of our tool. The results demonstrated a 92% positive feedback of our developed tool for ransomware analysis.

In future, more functionalities can be added to the proposed model, e.g., categorizing the ransomware family, increasing the efficiency of the tool.

**Funding Statement:** The authors acknowledge the support of Security Testing-Innovative Secured Systems Lab (ISSL) established at University of Engineering & Technology, Peshawar, Pakistan under the Higher Education Commission initiative of National Center for Cyber Security (Grant No. 2(1078)/HEC/M&E/2018/707).

**Conflicts of Interest:** The authors declare that they have no conflicts of interest to report regarding the present study.

## References

[1] R. Richardson and M. M. North, "Ransomware: Evolution, mitigation and prevention," *International Management Review*, vol. 13, no. 1, pp. 10–27, 2017.

[2] B. Jethva, "A new ransomware detection scheme based on tracking file signature and file entropy," Ph.D. dissertation. University of Victoria, Canada, 2019.

[3] B. Celiktas and E. Karacuha, "The ransomware detection and prevention tool design by using signature and anomaly based detection methods," Ph.D. dissertation. Istanbul Technical University, Turkey, 2018.

[4] J. A. H. Silva and M. Hernández-Alvarez, "Large scale ransomware detection by cognitive security," in *Proc. IEEE Second Ecuador Technical Chapters Meeting (ETCM)*, Salinas, California, USA, pp. 1–4, 2017.

[5] L. Rudman and B. Irwin, "Dridex: Analysis of the traffic and automatic generation of IOCS," in *Proc. Information Security for South Africa (ISSA)*, Johannesburg, pp. 77–84, 2016.

[6] H. Daku, P. Zavarsky and Y. Malik, "Behavioral-based classification and identification of ransomware variants using machine learning," in *Proc. 17th IEEE Int. Conf. On Trust, Security And Privacy In Computing and Communications/12th IEEE Int. Conf. On Big Data Science And Engineering (TrustCom/BigDataSE)*, New York, NY, USA, pp. 1560–1564, 2018.

[7] M. S. Kumar, J. Ben-Othman and K. G. Srinivasagan, "An investigation on wannacry ransomware and its detection," in *Proc. IEEE Sym. on Computers and Communications (ISCC)*, Natal, Brazil, pp. 1–6, 2018.

[8] A. Zahra and M. A. Shah, "IoT based ransomware growth rate evaluation and detection using command and control blacklisting," in *Proc. 23rd Int. Conf. on Automation and Computing (ICAC)*, Huddersfield, UK, pp. 1–6, 2017.

[9] B. Lokuketagoda, M. P. Weerakoon, U. M. Kuruppu, A. N. Senarathne and K. Y. Abeywardena, "R-killer: An email based ransomware protection tool," in *Proc. 13th Int. Conf. on Computer Science & Education (ICCSE)*, Colombo, Sri Lanka, pp. 1–7, 2018.

[10] N. Andronio, S. Zanero and F. Maggi, "Heldroid: Dissecting and detecting mobile ransomware," in *Proc. 18th Int. Sym. on Recent Advances in Intrusion Detection*, Kyoto, Japan, pp. 382–404, 2015.

[11] B. A. S. Al-rimy, M. A. Maarof and S. Z. M. Shaid, "Ransomware threat success factors, taxonomy, and countermeasures: A survey and research directions," *Computers & Security*, vol. 74, no. 9, pp. 144–166, 2018.

[12] I. Kara and M. Aydos, "Static and dynamic analysis of third generation cerber ransomware," in *Proc. Int. Congress on Big Data, Deep Learning and Fighting Cyber Terrorism (IBIGDELFT)*, ANKARA, Turkey, pp. 12–17, 2018.

[13] Q. Chen and R. A. Bridges, "Automated behavioral analysis of malware: A case study of wannacry ransomware," in *Proc. 16th IEEE Int. Conf. on Machine Learning and Applications (ICMLA)*, Cancun, Mexico, pp. 454–460, 2017.

[14] R. Brewer, "Ransomware attacks: Detection, prevention and cure," *Network Security*, vol. 2016, no. 9, pp. 5–9, 2016.

[15] S. S. Hansen, T. M. T. Larsen, M. Stevanovic and J. M. Pedersen, "An approach for detection and family classification of malware based on behavioral analysis," in *Proc. Int. Conf. on Computing, Networking and Communications (ICNC)*, Kauai, Hawaii, USA, pp. 1–5, 2016.

[16] A. Gazet, "Comparative analysis of various ransomware virii," *Journal in Computer Virology*, vol. 6, no. 1, pp. 77–90, 2010.

[17] C. Moore, "Detecting Ransomware with Honeypot Techniques," in *Proc. Cybersecurity and Cyberforensics Conf. (CCC)*, Amman, Jordan, pp. 77–81, 2016.

[18] D. Nieuwenhuizen, "A behavioural-based approach to ransomware detection," *Whitepaper. MWR Labs*, pp. 1–18, 2017.

[19] A. Arabo, R. Dijoux, T. Poulain and G. Chevalier, "Detecting Ransomware Using Process Behavior Analysis," *Proc. Computer Science*, vol. 168, no. 14, pp. 289–296, 2020.

[20] S. Jan, O. B. Tauqeer, F. Q. Khan, G. Tsaramirsis, A. Ahmad *et al.,* "A framework for systematic classification of assets for security testing," *Computers, Materials & Continua*, vol. 66, no. 1, pp. 631–645, 2021.

[21] C. Sandbox, "Cuckoo Sandbox—Automated Malware Analysis," 2017. [Online]. Available: http://www.cuckoosandbox.org.