International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-9 Issue-2, December, 2019

# Crypto Ransomware Detection on Windows Operating System

**Wira Z. A. Zakaria, Mohd Faizal Abdullah, Othman Mohd, Aswami Ariffin, Ng Thiam Tet**

***Abstract:*** ***Crypto-ransomware is a kind of malware threat, and it is one of approach frequently used by cybercriminals. It is due to the capability to hijack the victim's files and data by totally encrypting it using sophisticated cryptographic libraries such as OpenSSL and Microsoft Cryptography API. From the ransom note left by the attacker on the infected machine, the victim is told to fulfil the requested payment to get back the files. New variants of ransomware were released from time to time, thus making the task of detecting and analyzing it becomes challenging and resource consuming. Obfuscation and polymorphism employed in most modern malware made the task of identifying it even harder. This research investigates the domain of detecting ransomware on a Windows-based platform. We reviewed some of the related works done within this domain. In this research work, we proposed a framework for crypto-ransomware detection on the Windows-based platform by using information such as API calls and registry.***

***Index Terms:*** ***Crypto ransomware, ransomware, ransomware classification, Windows ransomware detection***

## I. INTRODUCTION

Ransomware is a type of malicious software (malware) that locks the system or encrypts the victim's files and demands an amount of ransom to get the files, data, and the system restored [1]. Same as any other malware, ransomware also has its technique to infect the victim machine. Ransomware utilizes multiple attack vectors such as social engineering, spam emails, botnet, evading detections and self-propagating through vulnerabilities [2], [3]. After it successfully infected the victim's devices, it will lock the files, folder or even the whole machine and encrypt the files that have the targeted extensions such as *.docx, *.xslx and *.jpg, to name a few. The endgame here is the victim will become unable to access the files or system until the victim fulfils the requested ransom to the attacker within a specific time frame [2].

Ransomware is divided into two categories, crypto and locker.

Revised Manuscript Received on December 30, 2019.
* Correspondence Author
**Wira Z. A. Zakaria,** MyCERT, Cybersecurity Malaysia, Cyberjaya, Selangor, Malaysia.
**Mohd Faizal Abdollah,** Faculty of Information and Communication Technology, Universiti Teknikal Malaysia, Melaka, Hang Tuah Jaya, Melaka, Malaysia.
**Othman Mohd,** Faculty of Information and Communication Technology, Universiti Teknikal Malaysia, Melaka, Hang Tuah Jaya, Melaka, Malaysia.
**Aswami Ariffin,** Cybersecurity Malaysia, Cyberjaya, Selangor, Malaysia.
**Ng Thiam Tet,** Faculty of Information and Communication Technology, Universiti Teknikal Malaysia, Melaka, Hang Tuah Jaya, Melaka, Malaysia



Crypto-ransomware is designed to encrypt the files that contain the targeted extensions. It is usually created by using popular production applications. For example, Microsoft Office, databases, picture and video editing applications. These files are assumed to have high personal and production values, either for the individual or organization. By encrypting all the high-value production files, the ransomware infection denies the system owner from accessing their own files and data. Crypto-ransomware makes use of the public-private key relationship, in which files are encrypted using the public key, and the private key is used to decrypt the files[1], [2].

After the encryption process ended and all the files that contain the targeted extensionsare encrypted, a ransom note is displayed on the screen, informing the victim about the file encryption and the payment instructions. The ransom note is shown either by changing the desktop wallpaper or popping up a new window. The ransom note contains the instructions on how to make payment to regain access to the files. The attacker will only provide the decryption key for the files if the victim pays the requested ransom within the specified time frame determined by the attacker. Anyway, by paying the ransom is not a confirmation that the files will be recovered.

In contrast, locker ransomware is not as disastrous as crypto-ransomware. It only locks out the system owner from the platform and it does not do any direct operations towards the files [4]. However, it also demands some ransom to be paid to restore access to the system or device.

## II. PHASES OF A RANSOMWARE ATTACK

### A. Dissemination

The most popularapproach of spreading ransomware is through phishing e-mail. Social engineering skill is used to write the email with the purpose to lure the victim into downloading and executing the attached malicious program. Some examples of the attachments are executable files and Microsoft Office files with macros. Another approach of spreading ransomware is through malicious webpages and exploit kit (EK) such as Angler EK.

### B. Installation

The infection processes started after the malicious payload has copied itself onto the victim's computer. At this stage, the malicious program is automatically installed and added new entries in the Windows registry to maintainits persistenceon each system reboot.

### C. Command & Control (C&C)

After the ransomware finished with the installation phase, the ransomware makes interactions with the command and control server. The objective here is to obtain the encryption key and additional instructions. This communication is varying between all ransomware families.

### D. Encryption

All the files with targeted extensions will be encrypted by the ransomware based on the key supplied by the command and control server. The encrypted file will be retained on the computer and the original ones will be deleted.

### E. Blackmail

After the all files are encrypted, the victim is prompted with a ransom note. The note is contained within a window, with a list of instructions for the victim to make the ransom payment in order to get the decryption key for all the encrypted files. Figure 1 below shows a summary of the five phases in a ransomware attack.

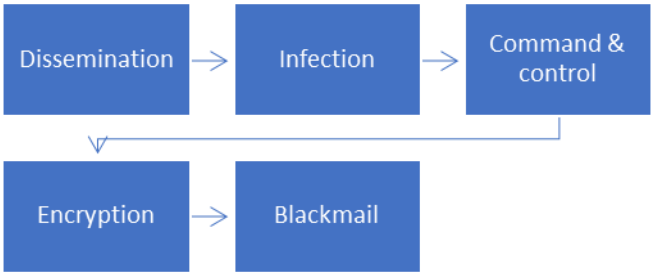


**Fig. 1 Five stages of a ransomware attack**


**Fig. 2 An example ransom note left by Badrabbit ransomware**

## III. LOCKER RANSOMWARE

This category of ransomware only locks the device without modifying the files contained within the device. Itprevents the victim from accessingit and its system functionalities[1], [6]–[8].

Locker ransomware is created with the purpose to prevent user access to the computing device or platform. Most of the time, it locks the device's graphical user interface (GUI), showing a ransom note and then requesting the device's owner to pay some money to restore access. Even though the device is locked, the ransomware purposely left it with minimal capabilities, just to only allowing the victim to pay the requested ransom[9].

This type of ransomware is designed to prevent access to the computing platform interface, and it does not make any changes to the files and data. System administrators and victims that are tech-savvy could find workaround to remove the ransomware and restore the system to its original state [9].

## IV. CRYPTO RANSOMWARE

This type of ransomware is far more dangerous than locker. It is the most preferred ransomware used by cyber criminals to get a lucrative income by extorting money from the victims. Cryptographic ransomware or usually known as crypto-ransomware involves encryption of files in its attack. After it managed to infect a computing host, it will search for all files that has specific extensions and encrypts all of it, either it is located on the local storage or even shared network drives [7]. Most crypto-ransomware utilizes the public-private key approach to encrypt the victim's files. Using the public key, data is encrypted. The private key, meanwhile, is used to decrypt the data [8]. The victim will be unable to access the documents and data by encrypting the files.

The victim is notified with a ransom note that contains a message about what has happened to the files on the platform and provided instructions on how to make the ransom payment to obtain the decryption key for all the encrypted files [6], [10]. The ransom payment is done using e-currency method such as Bitcoin. The main reason is to ensure that the communications for the transactions of the ransom payment between the victim and the attacker remain untraceable [3], [8], [11]–[13].

**Table. 1 List of some of the ransomware since 1989 to 2018**

| Ransomware | Type |
|---|---|
| AIDS Trojan | Locker ransomware |
| Reveton | Locker ransomware |
| CryptoLocker | Crypto ransomware |
| CryptoWall | Crypto ransomware |
| TeslaCrypt | Crypto ransomware |
| SamSam | Crypto ransomware |
| Locky | Crypto ransomware |
| Petya | Locker ransomware |
| WannaCry | Crypto ransomware |
| Badrabbit | Crypto ransomware |
| Notpetya | Crypto ransomware |
| Cerber | Crypto ransomware |
| Gandcrab | Crypto ransomware |

## V. RANSOMWARE DETECTION APPROACHES

Ransomware detection is an interesting area of research. Existing ransomware detection approaches are divided into two approaches: signature-based and behavior-based detection, as shown in Figure 3 below.

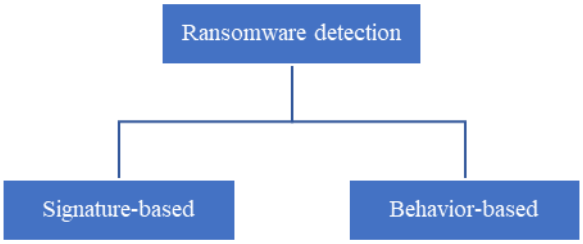


**Fig. 3 Approaches for ransomware identification**

Signature-based detection is widely used in anti-virus and intrusion detection system deployments. Early signature-based detection systems used a variety of features to detect malicious code. A signature is a byte sequence unique to a malware, which can be used to identify the specific malware. By using signature, any known malware can be identified by comparing its signature with the signature list in the antivirus database.This is achieved by searching for unique signatures in the database using a variety of pattern matching algorithm. Due to its low false positive, simple implementation and speed, most of antivirus solutions are using signature-based detection approach [23].

Signature-based anti-virus software will maintain and update periodically a list of signatures of known ransomware. The disadvantage of signature-based detection is it cannot identify unknown ransomware because its signature is not present in the signature database yet.

Detection method based on behavior makes full use of the actions performed during execution by the ransomware. The behavior of malware and benign binaries is analyzed during the training phase in behavior-based detection systems [23]. The ransomware and benign samples are run inside a malware sandbox and its behavior are captured in logs. Specific features such as API calls, system registries, file changes and so on is used to train and build a detection model. This model will be able to identify which one is malicious or benign. Some examples of machine learning algorithm that can be used to build this behavioral detection model k-nearest neighbor (kNN), Support Vector Machine (SVM) and Decision Tree.

## VI. CHALLENGES IN RANSOMWARE DETECTION

New and unknown variants of ransomware are released daily and signature-based detection tools such as antivirus are struggling to cope up. Signature-based anti-virus (AV) solutions detect ransomware based on their unique signatures. The major drawback of such solutions is that it has no knowledge of unknown malware. It will remain undetected until their signatures are present in the AV's signature database.

Another issue is in signature-based detection approach used by current AV solutions, since it cannot detect unknown malwares due to their polymorphic and metamorphic behavior [14]. As in ransomware case, most of the time, new variants changed the way it encrypts files and what happens to the data after it is encrypted. This make detection even harder.

Techniques such as code obfuscation and polymorphism, inherited from common malwares, signature-based detection system fails to detect new and unknown ransomwares. Ransomware developers never giving up in finding new ways to evade being detected. Hence, it is very important to develop new detection methods to detect the presence of ransomware on computing platform.

To address this issue, researchers are working towards finding patterns or features that represents the characteristics of ransomware activity on an infected computing platform.

## VII. TECHNIQUES FOR RANSOMWARE DETECTION

Ransomware detection is about detecting the presence of ransomware in the system. Detecting ransomware is a difficult task. Identifying it even before it begins the encryption stage is even harder; the current approach in ransomware detection highly depends on the signature [15]. If the signature of the ransomware is not present in the database of anti-malware, it will not be identified.

Existing approaches to detection of ransomware include techniques for detection of ransomware based on signature and behavior.A detection system can consider static features (e.g. byte entropy, program executable (PE) imports, and ASCII printable strings) to recognize malware, and a dynamic analysis system usually focuses on Windows API calls or network activity in the application.

Although static features may be useful in characterizing samples of malware, attackers can easily blur the malware code to confuse static analysis. However, in order to identify malicious activity, most ransomware behavior detection solutions rely on file system and registry events. Many ransomware detection solutions depend on application dynamic behaviors like changes in the registry and file system activities to identify malicious applications [16].

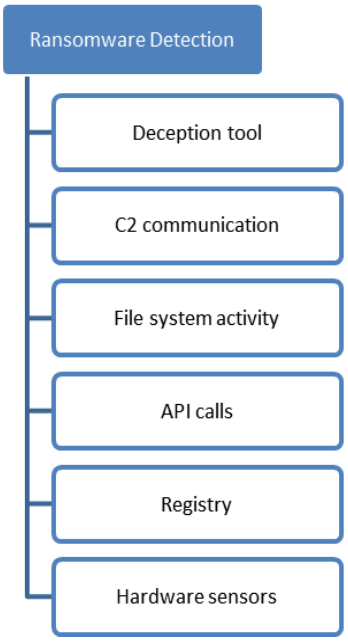


**Fig. 4 Shows a list of sources that can be used to identify the presence of ransomware on a Windows host**

### A. Deception tool

Honeypot is a computer system that is specifically deployed to detect unauthorized use of a resource. A honeypot system does not assume any valid interactions, so any communication observed by the honeypot is considered to be an attack, probe or compromise towards the system. This data will be used as an indicator for increasing an attack alert [15], [17].

In the target area, a set of fake files called 'honeyfiles' are deployed to lure and attract the ransomware [18]. Honeyfiles are set to act like FIFO rather than being standard archives, so the ransomware is blocked once it starts reading the file. The honeyfile solution will also automatically launch counter measures to stop the infection in addition to frustrate its behavior.

The solution framework does not need prior training or knowledge; hence, this method allows the identification of unknown, zero-day ransomware-related attacks if the ransomware interacts with deployed honeyfiles [18].

### B. Command and Control network communication

Another group of researchers used an SDN approach to identify ransomware activity by utilizing deep packet inspection to track the packet lengths of HTTP POST messages. Once ransomware is identified, the command and control (C2) server's IP addresses will be identified and blocked [4], [19].

### C. Filesystem activity monitoring

A real tool developed to detect file system activity-based crypto ransomware is UNVEIL [2]. Firstly, by generating a set of documents and adding them to the sandbox filesystem, it randomly generates a realistic user environment. Then UNVEIL extracts features from I/O requests such as request type (e.g. open, read, write) and data buffer entropy when present.These events are then matched against a set of I/O signatures on the access pattern as proof that the sample is actually ransomware [18].

Another researcher suggested a ransomware early warning detection system that monitors all file activities and warnings the user to something suspicious by monitoring a combination of three features; changes in file type, similarity measurement and entropy [20].

### D. API calls

For Windows platform, based on detail investigations of most cases, ransomware-specific events and processes are heavily related to Application Programming Interface (API) calls. This is because, user-level malware like ransomware, requires the invocation of system calls to interact with the operating system (OS) to execute its malicious actions [21].

Application Programming Interface (API) calls are the functions that a program utilizes in its execution. In other words, API calls are a set of routines provided by the OS for build applications, in which each of the API call performs a specific task [1]. APIs helps software programmers to code with ease with the usage of a stable and portable interface without the need to rewrite lower level functions [1]. As for Win32 APIs or Windows APIs, it can be separated into the common three modes which are User-mode, Native-mode and Kernel-mode [22].

The API calls list is extracted from a binary executable by static ransomware binary analysis with disassembly tools like IDA Pro or through dynamic analysis after running the ransomware sample in a sandbox environment like Cuckoo Sandbox [23].

For example, by creating a new one and making it persistent a significant number of locker ransomware samples use functions such as Create Desktop to lock the victim's desktop.In the case of crypto ransomware (e.g. CryptoWall), it is common to use standard system functions such as CryptEncrypt to encrypt files. Regretfully, attackers can easily bypass this by developing their own cryptosystems [18].

### E. Windows registry monitoring

The Windows Registry is a hierarchal database which has been utilized in older Microsoft Operating System since Microsoft Windows 98 all the way up until the latest OS which is Windows 10. The said database's functionality is to keep information that is required to configure the system, application and hardware devices. The Registry is used as a replacement for the older configuration files (config.sys, autoexec.bat, win.ini, system.ini) that are found in MS-DOS and Windows 3.x operating system [24]. Even though all current Windows operating systems have the Registry, there are some differences among them. Malware aims to change Windows operating system and application software by utilizing the registry [25].

In most cases, ransomware modifies the value in the Windows registry, for example to maintain the ransomware persistence during system reboot. This is the case of HKEY_LOCAL_MACHINE\Software\Microsoft\ Windows\CurrentVersion\Run for the automatic execution of programs at the session start [18], [26].

### F. Onboard hardware sensors

Data collected from hardware sensors present embedded in modern computing systems are monitored and appropriate criteria are used that enable the sensor data to detect the presence of ransomware infections. Detection of encryption depends on the use of small but differentiated changes in a system's physical state as reported by on-board sensor readings. A feature vector is formulated that consists of different sensor outputs, coupled with a detection criterion for the ransomware binary state present versus normal operation. An advantage of this approach is that previously unknown or 0-day ransomwares are vulnerable to this method of detection since it requires no prior knowledge of the malware, such as its signature, to deploy and use this method [27].

## VIII. PROPOSED FRAMEWORK

To detect the presence of ransomware on Windows-based end point, we proposed a detection system that consists of multiple monitoring modules and the output is a prototype for ransomware detection on Windows platform. The monitoring modules are inspired by the list of sources of possible

ransomware indicators on a Windows platform listed in the previous section. The modules involve in this proposed system is listed below:

### A. Windows Registry Monitoring Module

In malware analysis, utilizing registry offers valuable information to help understand thefollowings: changes by specific programs, signs of theinfected computer, and artefact of persistence mechanisms.

### B. Folder Monitoring Module

Monitors the folders for unidentified files and other binaries.

### C. Process Monitoring Module

Process is the direct indicator of running includes the malware. The observation of process spawning process tree, the parameters, image path, and loaded DLLs.

### D. File Monitoring Module

The increase or decrease in files indicates that ransomware may drop files related to ransomware, modify specific files, or remove artifacts generated by ransomware in order to hide themselves. For instance, encryption of user files will result in frequent I/O operations of the file system.

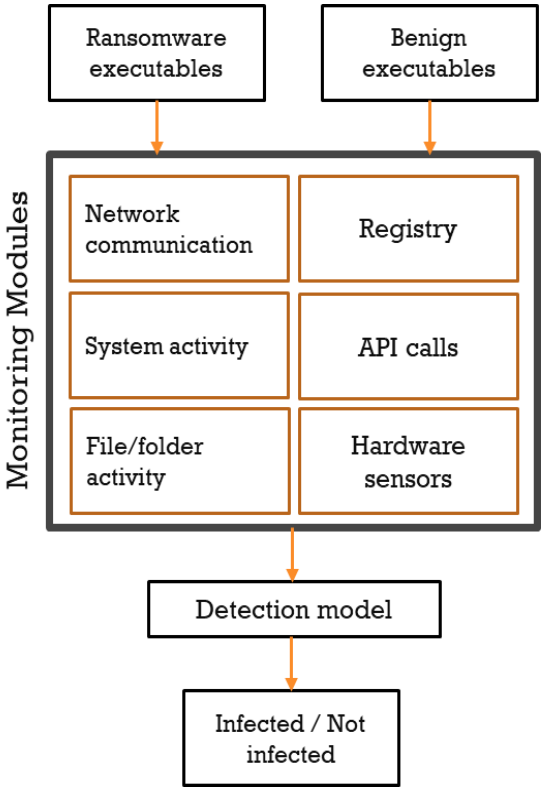


**Fig. 5 Architecture of the proposed system**

## IX. RESULT

We ran 435 ransomware samples using the Cuckoo malware sandbox setup. We managed to collect hundreds of ransomware communication and behavioral analysis logs in Javascript Object Notation (JSON) format. The logs contain textual and numerical information that is needed for this research to proceed with the next stage which is feature extraction and feature selection in order to train and build a machine learning model for ransomware detection.

## X. CONCLUSION

With the many variants produced on daily basis, ransomware detection is a resource consuming task. There is a need to build better and faster ransomware detection mechanism to prevent it from bringing more damage to our IT resources. In future, we will further this research by developing a supervised machine learning model and train it with ransomware behaviors data collected from the malware analysis sandbox.

## REFERENCES

1. S. Maniath, A. Ashok, P. Poornachandran, V. G. Sujadevi, A. U. P.Sankar, and S. Jan, “Deep learning LSTM based ransomware detection,” 2017 Recent Dev. Control. Autom. Power Eng. RDCAPE 2017, vol. 3, pp. 442–446, 2018.
2. A. Kharaz, S. Arshad, C. Mulliner, W. Robertson, C. Mulliner, and W. Robertson, “UNVEIL: A Large-Scale, Automated Approach to Detecting Ransomware,” 2016.
3. S. Homayoun, A. Dehghantanha, M. Ahmadzadeh, S. Hashemi, and R. Khayami, “Know Abnormal, Find Evil: Frequent Pattern Mining for Ransomware Threat Hunting and Intelligence,” IEEE Trans. Emerg. Top. Comput., vol. 6750, no. c, pp. 1–1, 2017.
4. G. Cusack, O. Michel, and E. Keller, “Machine Learning-Based Detection of Ransomware Using SDN,” 2018.
5. L. J. G. Villalba, A. L. S. Orozco, A. L. Vivar, E. A. A. Vega, and T.-H. Kim, “Ransomware Automatic Data Acquisition Tool,” IEEE Access, vol. 3536, no. c, pp. 1–1, 2018.
6. P. B. Pathak and Y. M. Nanded, “A Dangerous Trend of Cybercrime: Ransomware Growing Challenge,” Int. J. Adv. Res. Comput. Eng. Technol., vol. 5, no. 2, pp. 371–373, 2016.
7. Symantec, “Internet Security Threat Report,” Symantec, vol. 21, no. 2, pp. 1–3, 2016.
8. I. Yaqoob et al., “The rise of ransomware and emerging security challenges in the Internet of Things,” Comput. Networks, vol. 0, pp. 1–15, 2017.
9. K. Savage, P. Coogan, and H. Lau, “The Evolution of Ransomware,” Secur. Response, p. 57, 2015.
10. A. Continella, P. Di Milano, A. Guagnelli, and G. Zingaro, “ShieldFS: The Last Word In Ransomware Resilient Filesystems,” no. March 2014, 2015.
11. R. Brewer, “Ransomware attacks: detection, prevention and cure,” Netw. Secur., vol. 2016, no. 9, pp. 5–9, 2016.
12. D. Sgandurra, L. Muñoz-González, R. Mohsen, and E. C. Lupu, “Automated Dynamic Analysis of Ransomware: Benefits, Limitations and use for Detection,” 2016.
13. Z. A. Genç, G. Lenzini, and P. Y. A. Ryan, “The Cipher, the Random and the Ransom: A Survey on Current and Future Ransomware,” 2017.
14. A. Sami, B. Yadegari, N. Peiravian, S. Hashemi, and A. Hamze, “Malware detection based on mining API calls,” SAC ’10 Proc. 2010 ACM Symp. Appl. Comput., 2010.
15. C. Moore, “Detecting ransomware with honeypot techniques,” Proc. - 2016 CybersecurityCyberforensics Conf. CCC 2016, pp. 77–81, 2016.
16. S. Homayoun et al., “DRTHIS: Deep ransomware threat hunting and intelligence system at the fog layer,” Futur. Gener. Comput. Syst., vol. 90, pp. 94–104, 2019.
17. W. Z. A. Zakaria and M. L. M. Kiah, “A review of dynamic and intelligent honeypots,” ScienceAsia, vol. 39, no. SUPPL.1, 2013.
18. J. A. Gómez-Hernández, L. Álvarez-González, and P. García-Teodoro, “R-Locker: Thwarting ransomware action through a honeyfile-based approach,” Comput. Secur., vol. 73, pp. 389–398, 2018.
19. K. Cabaj, M. Gregorczyk, and W. Mazurczyk, “Software-Defined Networking-based Crypto Ransomware Detection Using HTTP Traffic Characteristics,” 2015.
20. N. Scaife, H. Carter, P. Traynor, and K. R. B. Butler, “CryptoLock (and Drop It): Stopping Ransomware Attacks on User Data,” Proc. - Int. Conf. Distrib. Comput. Syst., vol. 2016–Augus, pp. 303–312, 2016.
21. R. Veeramani and N. Rai, “Windows API based Malware Detection and Framework Analysis,” … Conf. Networks Cyber Secur., vol. 3, no. 3, pp. 1–6, 2012.

22. S. Z. MohdShaid and M. A. Maarof, “In memory detection of Windows API call hooking technique,” I4CT 2015 - 2015 2nd Int. Conf. Comput. Commun. Control Technol. Art Proceeding, no. August, pp. 294–298, 2015.
23. A. A. E. Elhadi, M. A. Maarof, and B. I. A. Barry, “Improving the detection of malware behaviour using simplified data dependent API call graph,” Int. J. Secur. its Appl., vol. 7, no. 5, pp. 29–42, 2013.
24. H. Carvey, “The Windows Registry as a forensic resource,” Digit. Investig., vol. 2, no. 3, pp. 201–205, 2005.
25. S. Romana, S. Phadnis, H. Pareek, and P. R. L. Eswari, “Behavioral malware detection expert system – tarantula,” vol. 196, no. July 2015, 2011.
26. Monika, P. Zavarsky, and D. Lindskog, “Experimental Analysis of Ransomware on Windows and Android Platforms: Evolution and Characterization,” ProcediaComput. Sci., vol. 94, pp. 465–472, 2016.
27. M. A. Taylor, K. N. Smith, and M. A. Thornton, “Sensor-based Ransomware Detection,” no. November, pp. 1–8, 2017.
28. L. Liu, B. Wang, and Q. Zhong, “Automatic malware classification and new malware detection using machine learning,” Front. Inf. Technol. Electron. Eng., pp. 1–26, 2015.
29. M. Sewak, “Comparison of Deep Learning and the Classical Machine Learning Algorithm for the Malware Detection,” 2018 19th IEEE/ACIS Int. Conf. Softw. Eng. Artif. Intell. Netw. Parallel/Distributed Comput., pp. 293–296, 2018.

## AUTHORS PROFILE

**Wira Z. A. Zakaria**, MyCERT, Cybersecurity Malaysia, Cyberjaya, Selangor, Malaysia.

**Mohd Faizal Abdollah**, Faculty of Information and Communication Technology, Universiti Teknikal Malaysia, Melaka, Hang Tuah Jaya, Melaka, Malaysia.

**Othman Mohd**, Faculty of Information and Communication Technology, Universiti Teknikal Malaysia, Melaka, Hang Tuah Jaya, Melaka, Malaysia.

**Aswami Ariffin**, Cybersecurity Malaysia, Cyberjaya, Selangor, Malaysia.

**Ng Thiam Tet**, Faculty of Information and Communication Technology, Universiti Teknikal Malaysia, Melaka, Hang Tuah Jaya, Melaka, Malaysia