Optimised Malware Detection in Digital Forensics

SaeedAlmarri and Dr Paul Sant

Institute for Research in Applicable Computing, University of Bedfordshire, Luton, United Kingdom

Associate Dean, University Campus Milton Keynes, Milton keynes, United Kingdom

Abstract

On the Internet, malware is one of the most serious threats to system security. Most complex issues and problems on any systems are caused by malware and spam. Networks and systems can be accessed and compromised by malware known as botnets, which compromise other systems through a coordinated attack. Such malware uses anti-forensic techniques to avoid detection and investigation. To prevent systems from the malicious activity of this malware, a new framework is required that aims to develop an optimised technique for malware detection. Hence, this paper demonstrates new approaches to perform malware analysis in forensic investigations and discusses how such a framework may be developed.

Keywords

Denial of service (DOS), Wireshark, Netstat, TCPView, The Sleuth Kit (TSK), Autopsy, Digital Forensics, Malware analysis, Framework

1. Introduction

Over the last decade, there have been noteworthy improvements in techniques to detect malware activities [1]. Loading and distributing executable files over the Internet always presents a risk to the overall security of the system [2]. Malware programmes can be installed by attaching hidden malicious code in an innocuous file or application. The code can then be activated by a remote programmer with the aim of threatening the existing system. According to a study by Islam et al. on the risk of downloading [3], of more than 450,000 files downloaded, approximately 18% contained malware programs. They also investigated whether different code investigation techniques yielded the same results. Astonishingly, they found that there were many cases where forensic investigatory tools were unable to detect the malware content of the infected files.

A significant amount of effort has been expended on developing techniques to perform robust computer forensic investigations [6]. Such effort has focused on collecting, analysing and preserving evidence of malware activities, for e.g. a study on botnets [4] and a study of executable spyware and client-sided honeypots[5] also illustrated defensive mechanism for securing a system both on the client and server side access. Other reports mentioned in [3][6] have also focused on acquiring large and diverse samples of malware to enable researchers and forensic experts to understand their nature and its rationale. Some existing tools like ERA remover, conficker, etc. can execute hidden and anonymous files and monitor their behaviour. These tools provide protection from all threats related to the malware functioning in the system. According to reports by Kasama et al (2012), a single piece of malware can compromise and infect the entire network system. Thus, protecting systems from unwanted malicious code can be considered as one of the most critical concerns in information security [6].

Various protocols and services have been implemented to simulate the behaviour of an end user in a compromised system, which assists researchers to analyse and relate malware behaviour in order to develop a better detection tool. However, different encryption techniques and complex programmes make it difficult to detect the activity of malware [1][3]. Commercially, there are various tools existing, but they are not being used to detect in an optimal way. Hence, this makes malware a serious threat to the domain of information technology and associated services [7].

1.1 Problem Statement

Malware is an intended program that performs malicious activity. It can be easily installed in any computer with a malicious intent. In a most apt scenario [2], malware attacks by organised crime syndicates largely target financial institutes and the banking industry, where they attack on their online software services that deal with the system of monetary assets ownership. Cybercriminals also enable malware attacks on customers and businesses dealing with financial institutions [2].

Bradford &Yegneswaran (2007) mentioned their report on the increase in malware activity through the use of sophisticated tools and methods, making it complicated for its detection [8].Also in a study by SANS institute (2012) illustration on the challenges in digital investigation on malware was mentioned. According to the report, gathering and analysing large chunks of data, obfuscated malware or malicious code are usually time consuming, costly and requires different techniques The situation becomes even more complex when the same data are present over a large network or various computational systems. This makes its challenging for an investigator to gather evidence of illegal distribution with pre-existing forensic tools, pre-defined malware detection systems and a limited duration at a workstation [9].

In cases where numerous systems or parties are involved in the same crime, the analysis of independent digital data during the investigation can result in the loss of indispensable correlated evidence. The loss occurs due to the incapability of digital forensics to craft correlations between multiple malware cases. It is exceptionally difficult to detect malware and to obtain accurate data due to numerous obfuscation techniques used by the programmers [10].

As mentioned in [10], this problem can be addressed by focusing on the various digital forensics techniques currently used to extract, gather and analyse malware-infected systems. A new correlational technique that focuses on typical features of malware activity and that is capable of accumulating sufficient forensic evidence against the perpetrator and formulating it to produce against the perpetrator in the court is needed. The goal of our research is to optimise the research paradigm for malware analysis and to improve the investigatory experience by employing an active approach to detect malware programmes. The next section of the report explains the existing techniques for malware detection. With this in mind, we focus on developing a robust framework for malware analysis and detection.

2. Malware Detection and Digital Forensics

Digital forensics and malware detection exhibit many similarities. Both involve techniques for deep and extensive data mining. Comparisons between the two can be exploited to collect evidence on any incident related to malware activities. Thus a similarity-based technique is required to make the process of detection much quicker and easier for an investigator to get results [11]. In contrast to digital forensics, studies on malware detection aim only to find pathways and traces of malware activity and not the operation of the malware in a live system. The first step should consider the technical principles of digital forensics, as it can formulate elements to construct the research base of the investigation. Furthermore, it helps in robust explorations of the traces of different types of malware present in the system [11].

2.1. The digital forensics process

The digital forensics process, which was introduced in 2001–2002 at the SANS Institute, includes specific categories that begin with the identification phase and end with the final decision stage [12]. The different categories in digital forensics are illustrated in Figure 1.

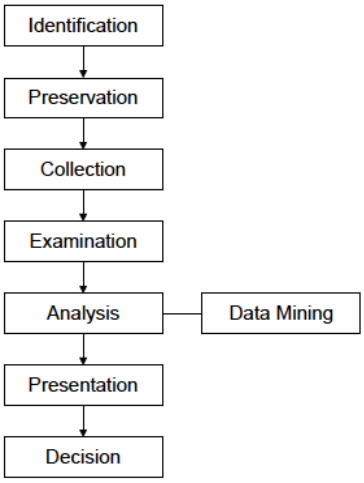

Figure1.Various categories in a digital forensics process

2.2. File system analysis

The file system refers to the organisation of various records contained in a system's database model. An analysis of the file system is a common process in digital forensics, as it provides valuable hidden information about system accessibility [13]. At the stage of data collection, the file system analysis extracts distinctive layers of data from the hard disk for later examination. Figure 2 illustrates the layers of abstraction by Farmer and Venema [13]. It shows three layers: (a) the user and the application view, (b) the file system view and (c) the hardware view. The data obtained from these layers can be combined to discover unseen, obfuscated malicious files. The process is advantageous for malware analysis in a case where data is kept hidden or encrypted behind the data files.

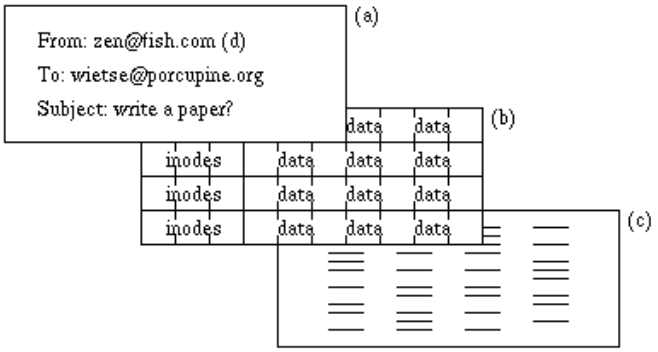

Figure 2. Various distinctive layers of abstraction [13]

2.3. Characteristics of malware

In both digital forensics and malware detection, it is vital to acquire sufficient data and definitions to make the detection system more efficient and the malware easier to recognise. Based on knowledge obtained from analyses of various malware domains, it is possible to recognise different patterns of malware, thus allowing an investigator to build a profile of the case. Different techniques and methods for malware detection proposed in [14][15] can be used to develop a new framework for a robust system. These techniques include methods such as time stamping, entropy analysis (File type, hashing, etc.) and obtaining traces from keywords and identifiers.

- **Time stamping:** This method includes an analysis of the duration between the attack and the initial phase of the investigation of the malware infection.
- **Entropy analysis:** The most commonly infected files are executable files. In some cases, the infection of these files is very difficult to detect. The infection may be hidden in the indexes, recycle bins or system folders. An example of this is Windows/system32.
- **Keywords and identifiers:** The simplest way of identifying malware code is via keywords, called 'identifiers'. These identifier scan be used to obtain data from IP addresses, email addresses and other sources to get the required information mentioned in a communication pattern. For example, 'key logger' can be a string used to find hints regarding any malware attack on the system.

3. Approaches to Malware Detection and Analysis

In digital forensics, any investigation includes two types of functions: 'D', for malware detector, and 'P', for the programme to be investigated. The maximum range for the D function relies on the range of programme P, which is defined as SET {malicious, genial} [16].

The function of detector D is to scan the programme P to check whether the P is malicious or genial. D detects the malware on the basis of the machine code, which stores the particulars, or the 'signature', of the malware. The investigation may yield one of three outcomes: false positive, false negative or the hit ratio [16].

- **False positive:** A false positive is when a scanner detects malware in a non-infected file. This means that the characteristics of a particular malicious code are not unique and hence the code appears in a non-infected file.
- **False negative:** This occurs when a detector fails to expose a particular malicious code in an infected domain. This usually happens when a scanner lacks a signature sample of a particular malware and hence fails to detect the infection.
- **Hit ratio:** The hit ratio occurs when the detector detects the malware accurately. This is possible only when the signature of a particular malicious code matches that of a stored sample in the detector's database.

Digital forensics distinguishes two types of malware detection methods: signature-based detection methods and anomaly-based methods.

3.1. Signature-based detection and analysis

Signatures are combinations of bytes, which are a constituent part of malware, or a malicious code. Malware is categorised into three forms: basic, metamorphic and polymorphic. Polymorphic is the most common. As this type of malware hides its identity and source of

generation, it becomes very hard for investigators to collect evidence against the plotter [16]. In basic malware, the source, or the 'initial point', of the programme is reformed, and complete control of the system is transmitted to the malicious payload. In polymorphic malware, the codes are mutated, while the original data in the system remain intact. This type of malware virus is replicated via a model-based engine, and thus the virus mutates every time the programme is executed in the same system. Metamorphic malware is based on reprogramming, which modulates the characteristics of the parent programme. This produces a new signature for each child variant that is created at a later stage [16].

Problems

- Extracting and dispensing the signature is very complex.
- Signature-based detection and analysis requires the investigator to perform penetrating research and analysis, which is only possible manually and in a controlled environment.
- Bypassing a signature during the investigation is a very common problem. Hence, new signatures should be updated every time before conducting the analysis.
- The ever-growing repositories of malware signatures are becoming very difficult to manage.

3.2. Specification-based detection and analysis [17]

This type of detection system depends on the particular specification of the implementation and the deployment of a particular application or system file. The process works by learning all phases of the application development and assuming that any abnormality carries malicious code. The process uses the concept of reverse engineering, which focuses on reaching the source of the programme. Sometimes it takes multiple runs to debug the source coding.

Problems

The main problem with this technique is its accuracy. The whole process depends on pre-defined rules to follow, which makes it hard to conduct practical analysis. Moreover, rules for the process need to be updated on a regular basis, as it is difficult for an investigator to learn any of it.

4. Requirements for the New Framework

Based on the above conditions faced by investigators, a new framework is required to optimise the results. This should be integrated into a system to detect and analyse malware efficiently and effectively. To ensure that analysis is comprehensive, it should perform both as a detector and a warning system. To understand the precision and the nature of the requirement, an extensive market research is required in the field to gather information for analysis.

4.1 Market Research

To find the precise problem statement for the research, qualitative and quantitative approaches were implemented using a questionnaire-based methodology. During the research, a questionnaire was distributed through email to respective respondents. The questionnaire used a framework based on a scale of items to assess data gathered from the respondents. A pilot study was accompanied to authenticate the questionnaire, where 30 completed questionnaires were collected. Around 43% of the respondents were professionals from IT security, 33% were forensics experts, 17% were network administrators and 7% belonged to other IT fields. Below are some of the sample questions:

1. How often you receive case studies related to Malware for investigation?				
<5%	5% to 20%	20% to 50%	50 to 80%	80% Above

2. How many of the above cases were solved with accurate results				
<5%	5% to 20%	20% to 50%	50 to 80%	80 Above

3. What is your opinion on level of skills and technological tools adopted in your company to carry investigating process on malware?		
Weak	Average	Strong

4. Do you think there is a need of a new customized tool for malware detection and analysis?		
Yes	No	Can't say

4.2 Discussion on the Result

The result from the observations reflects the vulnerabilities in the existing tools and methodologies present for malware detection. Although most of the experts were aware of the criticality and complexity of malware detection, still no specific solution for accurate detection was obtained. Some of the analysis report is explained in the following tables.

Table 1. Important factors from the analysis

Factors	Yes	No	Cant Say
Changes in Malware Landscape	52%	13%	35%
Organizational vulnerabilities	47%	20%	33%
Expertise on Malware	67%	27%	6%
Detection of malware by non-technical employees	17%	53%	30%

Table 2. Level of Skills in Investigation.

Weak	Average	Strong
13%	60%	25%

Table 3. Level of accuracy during investigation.

<5%	5%to 20%	20% to 50%	50 to 80 %	80 above
7%	20%	43%	27%	3%

Table 4. Choice of tools.

Open Source	Shareware	Commercial	Cant Say
7%	0%	43%	7%

Table 5. Need of new tool.

Yes	No	Cant Say
90%	3%	7%

Table 6. Preferable Methods.

Dynamic	Static	Other
57%	33%	10%

4.3. Result on Analysis

The results of the observations reflect the vulnerabilities in the existing tools and methodologies present for malware detection. Although most of the experts were aware of the criticality and complexity of malware detection, no specific solution for accurate detection was obtained. Some important factors from the analysis were identified that provide a clear view of the problem statement. The analysis is as follows:

- The core problem faced by the investigators was to find the level of malware infection in the file.
- Most importantly, investigators required using other parameters, such as Malware signature and behaviour in combination, for optimised detection.
- Most of the infections were both critical and complex. Thus, there is a need for a prerequisite method to reduce the associated complication.
- Although there are tools that are capable of detecting malware and other malicious code, most of them produce inaccurate or incomplete results.
- The cost of the tool played an important role in its adoption.
- Most of the investigators wanted to use tools based on both dynamic and static methodology.
- It usually took less time for the investigators to detect the presence of malware, but more time to investigate it forensically.
- Managing and updating the malware database is a task that should be given priority.

5. Proposed Framework

The required framework is designed for developing an optimised malware detection and analysis tool, which is based on the analysis of and the results obtained from the secondary research conducted in the survey report. The entire research pattern is divided into three distinctive phases. This distinctive framework defines the various policies, categorises the organisational assets and creates a new environmental variable for the process. The framework is designed for developing an optimised malware detection system that can confirm the malware infection using any standard of deployment. It can also assist in developing a robust and redefined metrics model that can work with a three-stage analysis covering all operational activities within the system. It is an unbiased model that initialises its operation by first confirming the malware infection, and then by analysing, detecting and proliferating it. The objective for developing this framework is to understand the concepts of the detection costs associated with dealing with malware discovery and analysis in an associated environment in order to provide accurate and precise information that can be used to increase the performance matrices. The entire process is divided into three phases:

Phase 1: Malware Acquisition Function,
Phase 2: Detection and Analysis
Phase 3: Database Operational Function.

Below is the diagram that defines the relationship between the different phases of the process. Each phase is explained by keeping the process of implementation in mind.

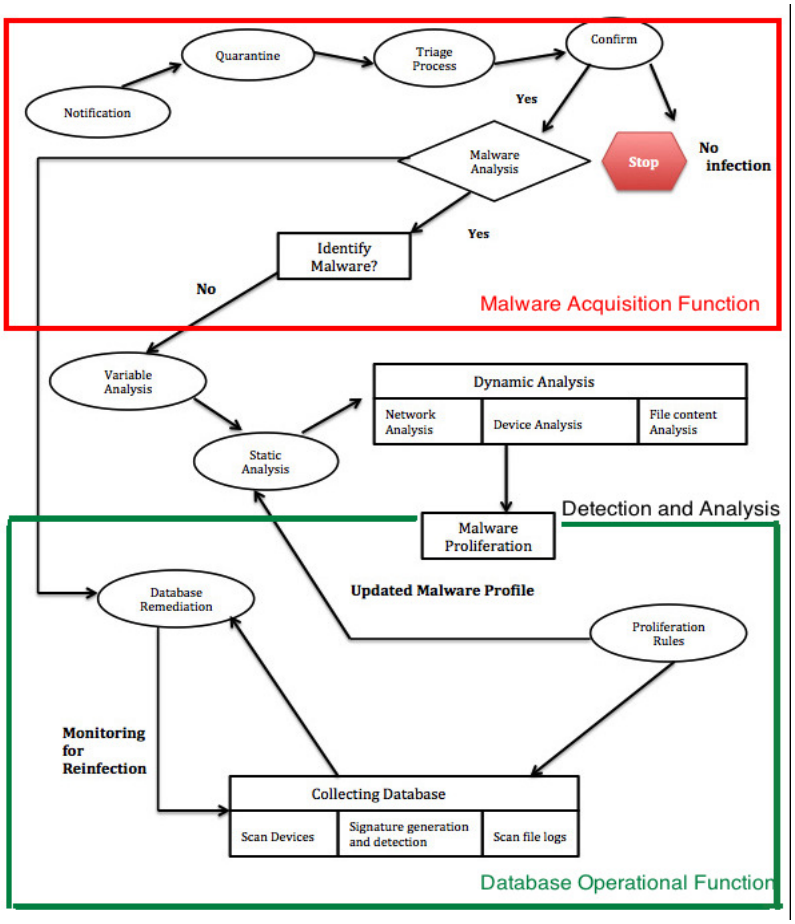

Figure 3. Three-phase process for malware detection and analysis

Phase 1: Malware acquisition function

Phase 1 initiates its task by confirming the occurrence of malicious code in the data. It deals with operations concerning the confirmation of the data that is infected by the malware. It also includes the functionalities that contain components that can detect both active and passive infections in the associated system. This phase includes the following components:

- **Notification:** Acquisition can begin in a variety of ways; including a notification alert sent from a third party (such as CMS-cash management systems or an online payment processor) or it can be an alert from an endpoint suit. It can act as a platform to determine whether or not the issues occurring in the system are realistic [21].
- **Quarantine:** The main objective of this component is to collect the damaged file from the system and then remove it separately from the rest of the associated files. This activity can prevent any duplication/replication of malicious code from being transferred into another non-infected file [21].
- **Triage process:** This process only works for identifying the critical nature of the issue that occurred due to the malware infection. It contains all sorts of analysis processes to categorise the problem by level of seriousness (high degree of seriousness or low degree of seriousness) [22].
- **Infection confirmation:** Up to this point in the process, we have obtained enough information to know whether or not the selected data is infected. So, now the detection tool

makes a decision and the next appropriate action is determined [22].

Depending on the information that is found during Phase 1, the following actions can occur:

- If the selected data is not infected, then the entire process can be stopped and things can revert to normal.
- If a malware infection is detected, we can decide whether to conduct an analysis of the selected malware infection or jump directly to the database remediation.
- If we decide to analyse the data, we must be sure to analyse it properly in order to obtain more information about the infection [23].
- If we identify the infection and do malware proliferation, we must check whether the selected infection matches up to the database; if it does not, then operations need to be performed to add the new information to update the existing malware profile [23].

Phase 2: Detection and analysis

Based on the results from the previous phase, it can now be confirmed that the files are infected by the malware. Therefore, the question arises as to how we can go about obtaining information about the activity that the phase was trying to perform on the system, whether the attack consisted of an attempt to steal a password or an attack on the confidentiality of the data. If it is difficult for an organisation to determine the exact level of threat, a complete investigation is required in order to fully analyse the infected file.

Detection and analysis constitutes functional output from the first phase (malware acquisition). After confirming the presence of a malware infection, the malicious code can be investigated and its characteristics can be analysed [25]. Defining the characteristics helps in proliferation, which can contribute to developing and updating the malware database (mentioned in Phase 3). The vital testing sub-functions include:

Build a control environment

Before doing any analysis, it is important to develop a control environment that can be used to perform a thorough investigation. This is mainly a one-time operative process, but with changes in the requirements, one must engage in an on-going analysis in order to achieve the target goals [24]. The following pre-defined guidelines are required:

- **Isolate the infected file**: Keep the infected file away from the live data. The infected file should be kept isolated in a separate network, as doing so can prevent a replication of the infection.
- **Provide a separate network**: During the investigation, it is important to have access to a separate Internet service provider system.
- **Testing tools**: Tools used for creating a previous testing environment should be cleaned perfectly before starting to work on them again.
- **Keep a log file**: A log file should be considered to be a special preference in the investigation, as it carries important information about the malware activities. Log files record each step that the malicious code used to infect the data file.
- **Sandbox:** It is always preferable to perform testing on a separate sandbox, as it becomes easy to conduct a specific investigation on the memory file that may contain a malicious infection.

Static Analysis

This function conducts a core analysis process on the suspected file. All results identified by this function can be added to the database carrying the malware proliferation function. Malware also

leaves evidence behind; thus, a thorough investigation of the executable files can be used to identify future threats to the system. Activities can be monitored internally in the system in a variety of ways, including network accessibility and performing operations on the file system and on the internal storage. To optimise the static analysis process, the following guidelines need to be followed (mentioned in [25]):

- **Initial analysis**: This includes investigating the obvious symptoms of attack in order to identify the nature of the infected file. First, we try to match up the fingerprint of the file to identify any existing malware profile in the database. This can usually be done using MD5 hashing.
- **Classifying the format of the file**: While analysing the Windows files, every executable file contains a schema, which may yield specific information about the cracker and its intention for the attack. Analysis of information such as version information, menus and calling functions can reveal a lot of evidence [25].
- **Analysing text strings**: It is always important to scan the isolated text strings that are present in the infected file. Using various keywords in the built-in search field can help in obtaining the associated evidence.
- **Debugging**: The last step for static analysis is to conduct a process of disassembly. This helps determine the patterns that can assist in identifying the attacker [25].

Dynamic analysis

As mentioned above, static analysis is an initial way to investigate malware in an infected file. We need to maintain the goal of creating a robust malware profile that can prevent all future malware attack attempts. Although malware infection can be detected at the earliest stage, one still may not be able to identify the unauthorised modulations that occurred and that compromised the system after the attack. So, here we also apply the dynamic analysis process. The dynamic analysis typically gathers information on the following (mentioned in [23]):

- **Memory**: Malware can bring changes to the buffer flow and can also tamper with the main memory of critical programmes. Thus, a deep analysis of the volatile memory can assist in obtaining good information about how/why malware activities were accessing internal memories.
- **Investigation registries**: This involves looking for the modifications that were done in the registry keys. The victimised machine can illustrate the exact changes that occurred in the machine, which can help the investigator detect easy changes in system.
- **Investigation processes and running services**: This can show all the processes that were either started or stopped intentionally by the malware. Analysing the processes illustrates some of the initial processes that can suggest the track record (origin) of the infected file in the system.
- **Looking for virtual machines**: Keeping an eye on the attached virtual machine (VM) can help an investigator continue the research process. Malware can go dormant after VM is detected, so the investigator is required to check for the presence of any VM on the hardware or a VM that is running dissolutely inside the system.

Dynamic analysis can be further differentiated into three types:

- Network analysis
- Device scanning
- File content analysis

While the static function can assist in analysing the features of executable files, the dynamic

function reads the metadata, which include information about the size of the file and its format and information about the library management system. These data can be accessed to detect malware in the infected file.

Phase 3: Database operational function

Database management is required to ensure that samples of the malware and information about the malware are available. This information can be used to update the profile of a particular malware, which can contribute to the process of performing an analysis and detecting the mechanism. Any failure in malware notifications (in Phase 1) can be remediated to the database operation in order to monitor the stage of reinfection of the malware in the system [26].

Malware profiling includes the following:

- Adding the aggregated information.
- Packaging the profile to make it more reliable for screening and scanning.
- Categorising the profiles so as to make it convenient for other investigators to access the information.
- Revisiting the malware profile periodically in order to prevent malware reinfection (an updated malware profile is another important function to consider).

6. Implementation of the Approach

By emphasising the described approaches in section 5, an experimental operation and evaluation can be undertaken to develop an optimised technique based on pre-defined correlational methodologies for robust malware detection. The following steps can be undertaken during the experiment:

6.1File content analysis

A file is an important part of the system, as it carries data processed by the user that demonstrates all activities done on the system before it was compromised. Hence, analysing file content can provide a major clue about the source and the characteristics of the malware.

One major problem faced by all previous detection techniques is the inability of the system to recognise obscure malware (e.g. malicious codes that are fully encrypted). In such cases, analysing the file content is always preferable compared to the previously mentioned approaches. File content analysis can be done either by calculating the entropy of the file or by analysing the particular string structure [15].

The entropy calculation method can be used to examine compressed files that are encrypted and hence modulated into different formats [15]. In the proposed scenario, the entropy calculation for any file type can be done with the ‘Ent tool’, which presents the difference in the sizes between an original file and an infected file in a tabular formation. String method analysis involves searching for particular keywords. Such analysis can be undertaken with the ICAT tool (from TSK), where TSK stands for The Sleuth Kit and Autopsy Browser (which acts as GUI for TSK) and can be employed to search for strings [18]

6.2Network analysis

A large amount of malware is distributed over the network to create redundancy and to ensure

effective propagation. An optimised network should be able to detect malware and other malicious codes. Trojan is the most commonly distributed malware. In this study, the netstat command and network packet analysis are discussed as follows:

- **Netstat command**

This performs the analysis on the TCP/IP network protocol. The conditions of the connection before and after deployment of malware on the system will provide a comparison result between the two. Deploying TCPView (for listing TCP and UDP endpoints) and Port Explorer (for exploring and analysing network sockets) can generate logs and records, which will be matched up with the original Listening PID to detect malware functioning [19].

- **Network packet analysis**

This technique investigates network traffic flowing on the network channels. Malicious code showing unusual activity enables evidence to be collected on the server side [20]. For performing the analysis, the Wireshark tool will be deployed on a virtual machine to explore the traffic on the suspected host. The main advantage of using a packet sniffer is to obtain knowledge about the DNS protocol packets and connections established between the machine and the IP address generator [20].

The proposed approach provides a robust framework for detecting malware. The following are its characteristics:

- It avoids allowing data chunks to accumulate, making it easier for investigators to flexibly detect malware without wasting much time.
- It focuses on distributed networks, as most other methodologies avoid getting into network data.
- With this approach, it is feasible to track the source of the malware, which can further prevent the chance of other malware attacks.

7. Conclusion

This paper discussed a new framework and approach for malware detection. Open source tools can be used in the extraction, investigation and analysis of the data. The report focuses on malware detection techniques used in digital forensics and data mining. The techniques mentioned in Section 4 can be used to perform experiments; hence, results pertinent to live systems will be generated. The overall outcome of the study is multi-fold, where the intention is to discuss various efficient and optimised techniques for use with malware detection. By recognising the faults in the existing systems, the new framework can overcome the limitations and thus can assist an investigator in obtaining evidence of malware to get an optimised result.

References

[1] Carrera,E. &Erdelyi. (2004). 'Digital genome mapping: Advanced binary malware analysis', Virus Bulletin, pp. 175-186.

[2] BITS (2011) 'Malware Risks and Mitigation Report' ITS/The Financial Services Roundtable 2011, Retrieved from http://www.nist.gov/itl/upload/BITS-Malware-Report-Jun2011.pdf

[3] Islam, N.,Anand, R., Jaeger, T.,&Rao, J.R. (2009). 'A flexible security system for using Internet content',Software, IEEE, Vol. 14, No.5, pp. 52,59. Retrieved from http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=605931&isnumber=13290

[4] Jehyun, L., Jonghun, K., Hyo, S., &Heejo, L. (2010). 'Tracking multiple C&C botnets by analysing DNS traffic.Secure Network Protocols (NPSec)',6th IEEE Workshop, pp.67, 72.

[5] Hayatle, O.,Otrok, H., & Youssef, A. (2012).'A game theoretic investigation for high interaction honeypots',Communications (ICC), 2012 IEEE International Conference, Vol. 10, pp.6662-6667.Retrieved from http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6364760&isnumber=6363631

[6] Kasama, T., Yoshioka, K., Inoue, D., & Matsumoto, L. (2012). 'Malware detection method by catching their random behaviour in multiple executions',Applications and the Internet (SAINT), IEEE/IPSJ 12th International Symposium, Vol. 2, pp.262,266. Retrieved from http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6305295&isnumber=6305234

[7] Mehdi S, Tanwani A, Farooq M (2009), "IMAD: in-execution malware analysis and detection", Proceeding of the 11th Annual conference on Genetic and evolutionary computation, pp: 1553-1560

[8] Barford,P.,&Yegneswaran,V (2007). An inside look at botnets. Special Workshop on Malware Detection,Advances in Information Security, IEEE Journal.

[9] Zahn K (2012), "Case Study: 2012 DC3 Digital Forensic Challenge Basic Malware Analysis Exercise", SANS Institute Reading Room site, Retrieved from https://www.sans.org/reading-room/whitepapers/malicious/case-study-2012-dc3-digital-forensic-challenge-basic-malware-analysis-exercise-34330

[10] Lim Y; Ryu H; Choi K; Park C; Park W; Kook K (2012) "A Study on Malware Detection System Model Based on Correlation Analysis Using Live Response Techniques," Information Science and Applications (ICISA), 2012 International Conference, vol., no., pp.1,6, 23-25

[11] Park J, Kim M, Noh B, Joshi J (2006), "A Similarity based Technique for Detecting Malicious Executable files for Computer Forensics", information Reuse and Integration, 2006 IEEE International Conference, vol., no., pp.188,193, 16-18

[12] Ryder, K. (2002). 'Computer forensics – We've had an incident, who do we get to investigate?'SANS Institute, GSEC Certification Assignment Version 1.3.

[13] Farmer, D. &Venema, W. (2004).Forensic Discovery.Addison Wesley Professional, USA

[14] Casey E (2002), "Error, Uncertainty, and Loss in Digital Evidence", International Journal of Digital Evidence Summer 2002, Volume 1, Issue 2, Retrieved from https://utica.edu/academic/institutes/ecii/publications/articles/A0472DF7-ADC9-7FDE-C80B5E5B306A85C4.pdf

[15] Davis T (2009), "Utilizing Entropy to identify Undetected Malware", White paper by Guidance Software, cyber security solution, Retrieved from http://image.lifeservant.com/siteuploadfiles/VSYM/99B5C5E7-8B46-4D14-A53EB8FD1CEEB2BC/43C34073-C29A-8FCE-4B653DBE35B934F7.pdf

[16] Neelakantan, S.&Rao, M. (2008). 'Threat-aware signature based intrusion-detection approach for obtaining network-specific useful alarms',Internet Monitoring and Protection. The Third International Conference, Vol. 2,No. 3, pp.80,85. Retrieved from http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4561330&isnumber=4561311

[17] Vinod P, Laxmi V, "Survey on Malware Detection Methods", Department of Computer Engineering, Malaviya National Institute of Technology, Retrieved from http://www.security.iitk.ac.in/contents/events/workshops/iitkhack09/papers/vinod.pdf

[18] Dowling A (2006), 'The Sleuth Kit v2.01 and Autopsy Forensic Browser Demonstration', White paper on TSK manual, Retrieved: http://www.sjones.co.nz/downloads/Files/Forensics/TSK_v201_Demonstration.pdf

[19] Vigna, G., & Kemmerer, R.A. (2008).'NetSTAT: A network-based intrusion detection approach', Computer Security Applications Conference, Vol. 7, No. 11, pp.25,34.

[20] Broadway, J., Turnbull, B., & Slay, J. (2008). 'Improving the analysis of lawfully intercepted network packet data captured for forensic analysis',Availability, Reliability and Security, 2008. ARES 08. Third International Conference, Vol. 4, No. 7, pp.1361,1368.

[21] Park, S. (2012).'Malware expert: Execution tracking', Cybercrime and Trustworthy Computing Workshop (CTC), 2012 Third, Vol. 48, No. 55, pp.29-30.

[22] Blount, J.J.,Tauritz, D.R., Mulder, S.A. (2011). 'Adaptive rule-based malware detection semploying learning classifier systems: A proof of concept', Computer Software and Applications Conference Workshops (COMPSACW), 2011 IEEE 35th Annual, Vol., No., pp. 110,115, 18-22.

[23] Adeel, M.&Tokarchuk, L.N. (2011). 'Analysis of mobile P2P malware detection framework through Cabir&Commwarriorfamilies', Privacy, security, risk and trust (passat), 2011 IEEE Third International Conference on SocialComputing (Socialcom), pp.1335,1343, 9-11.

[24] Chen, L., Liu, B.,Hu, H. &Zheng, Q. (2012). 'A layered malware detection model using VMM', Trust, Security and Privacy in Computing and Communications (TrustCom), 2012 IEEE 11th International Conference, pp.1259,1264, 25-27.

[25] Cavallaro L, Saxena P, R. Sekar (2008), "On the Limits of Information Flow Techniques for Malware Analysis and Containment", Computer Science Department University of California at Berkeley, Retrieved from http://www.comp.nus.edu.sg/~prateeks/papers/saxena-dimva08.pdf

[26] Moser, A.,Kruegel, C.&Kirda, E. (2007), 'Limits of static analysis for malware detection', Computer Security Applications Conference, 2007. ACSAC 2007. Twenty-Third Annual, Vol., No., pp. 21,430, 10-14.

Authors

SaeedAlmarri received his B.S.c in Information system from Ajman University of Science & Technology, College of Information Technology in 2008. He got his Master degree in Business information system from University of Bedfordshire, UK, in 2011. Currently, he is a PhD student at University of Bedfordshire with a thesis entitled " Malware Detection and Analysis".

Dr. Paul Sant completed his PhD from King's College, London in 2003 with a thesis entitled "Algorithmics of edge-colouring pairs of 3-regular trees" and prior to this, a BSc. in Computer Science from the University of Liverpool (1999). Paul is an active member of the British Computer Society and a Chartered Information Technology Professional (CITP) as well as being a fellow of the Higher Education Academy. In January 2013 Paul was appointed to a seconded position of Associate Dean working on a University Campus project. He still maintains strong links with the Department, and is still research attractive, being a local PI on the EU funded ECENTRE project.