CHAPTER 7: CONCLUSIONS AND FUTURE WORK

The key contribution of this research is the implementation of EGA worm detection and the response technique for cloud computing. The technique consists of various steps to integrate cloud worm behaviour analysis, classification, KDD process and GA for detection and response. A research gap was observed in cloud worm detection and analysis. This research shows a new state of art to overcome those gaps. The proposed method achieved better accuracy and performance compared with existing research.

This chapter concludes the findings of this research and shows a new pathway for future research direction. These directions lead towards continuing research in the cloud computing area. A detailed experimental evaluation is presented in this thesis. Experiment was conducted using Weka. Experimental results were compared with other established work.

7.1 Main Contributions

The main contributions of this research are as follows:

i) A new cloud worm classification technique is inspired by GA.

For worm detection, classification is one of the decisive processes that must be considered to ensure the effectiveness of the detection process. The classification method has been largely used in malicious code analysis specifically in measuring the efficiency of the detection for a new or unknown sample of malicious codes. This research proposed a new classification for cloud worm. The classification consists of five main features: infection, activation, payload, propagation and operating algorithm. Many sub features exist under these main features, as described in Chapter 4.

A categorised frequency analysis was undertaken after the classification. Analysis results indicate that approximately 60% of the infection was initiated through virtualisation. Approximately 14% of the infection was initiated by application. Approximately 42% worm was activated by human triggering in the

cloud. However, approximately 39 worms were activated by self-activation. The maximum payload of a worm is caused by registry shuffling which is approximately 21%. Approximately 43.9% of the worm operating algorithm is stealth. This attribute is a big thread for cloud users because of its personal account and bank account information stealing behaviour. Approximately 65.6% worm propagation is random which suggest that they do not follow any specific propagation method. This feature makes worm detection in the cloud extremely challenging.

After the categorised frequency analysis, the relationship between worm features was analysed statistically by employing Chi-square and symmetric measure. From the Chi-Square tests and symmetric measure, that relationship depends on the nature and the outcome or the action of sub features.

ii) Newly enhanced GA for cloud worm detection.

GA evolves a population of initial individuals to a population of high quality individuals wherein each individual represents a solution of the problem. Each individual is called a chromosome and is composed of a predetermined number of genes. The quality of each rule is measured by a fitness function as the quantitative representation of each rule's adaptation to a certain environment.

The procedure starts from an initial population of randomly generated individuals. The population is evolved for a number of generations while gradually improving the qualities of the individuals in the sense of increasing the fitness value as the measure of quality. During each generation, three basic genetic operators are sequentially applied to each individual with certain probabilities, namely, selection, crossover and mutation, as shown in Table 5.1, p. 128. The techniques are presented in Table 7.1.

Table 7.1: Summarisation EGA techniques.

Techniques	**EGA**
Selection	Selection Proportional of Fitness
Crossover	Tree Crossover
Mutation	Tree Mutation
Evolution	Evolution Controller

A GA named EGA is proposed from the motivation of OlexGA, and then integrated in Weka and finally evaluated. This work proposed GA parameters to improve detection accuracy.

The proposed techniques are used for selection, crossover, mutation and evolution. Four new algorithms are also proposed for these new techniques. For the experiments, 10 cross validation was used to evaluate the proposed techniques. For benchmarking, this work integrated OlexGA with Weka, which is also a GA. Experimental results were compared with OlexGA and other established classification algorithm, such as IBK, J48 and Naïve Bayes. The proposed GA with new techniques achieved 99.7% true positive, 1.4% false positive rate and 99.7% recall and F-measure rate. Approximately 99.749% was correctly classified by the proposed GA.

iii) A new rule for measuring cloud worm threat levels pertaining to weight and severity using CIA which was a proposed response action.

Security metrics was used in identifying the levels of risk to help determine the priority levels and corresponding actions. The rules of weight and severity value are justified by CIA. Weight is measured based on the security levels of these five main features: infection, activation, payload, propagation and operating algorithm. Weight value is defined based on CIA. Based on the weight value, severity value also defined and explained the worm threat levels as high,

medium and low. The response algorithm was proposed following these metrics. Details are provided in Chapter 6, Section 6.3, page no 171.

v) Enhanced worm technique for detection in cloud computing with 99.71 accuracy rate. Existing work have been compared, as presented in Table 5.5. The result is summarised in Table 7.2

Table 7.2: Experimental results various metric of various algorithms.

Algorithm	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Correctly Classified (%)
OlexGA	0.707	0.051	0.902	0.707	0.75	0.828	70.7113
EGA	0.997	0.014	0.997	0.997	0.997	0.992	99.749

The experiment conducted using the same dataset revealed that the accuracy rate of the OlexGA is low in terms of cloud worm classification. Therefore, this research improves the performance of GA algorithm as a classifier. GA is a search algorithm based on natural selection principles and genetics. GA generates new population from the existing population. Predicting the attack type that can handle unknown attacks in the future is possible by generating a new population. EGA was proposed in this chapter. This algorithm goes through security metrics, CIA, weight and severity measurements. EGA shows 99.74% accuracy rate with 0.997 true positive rates and 0.014 false positive rates with 10-fold validation. However, OlexGA algorithm shows 70.71% accuracy rate with 0.707 true positive rates and 0.051 false positive rates with 10-fold validation. The proposed EGA outperformed OlexGA.

The EGA response suggested isolating the infected host by turning it off. The system ignores a low threat level. The system is under monitoring if the threat level is medium. However, the system shuts down immediately if the threat level is high.

7.2 Future Research Direction

Despite the substantial contributions of the current thesis in EGA worm detection and response, a number of open research challenges must be addressed for the future advancement of the area. This research focused on worm attacks in the cloud on a Windows operating system. However, in cloud computing infrastructure, all hosts are based on Windows, and they could be in other operating systems (OS), such as UNIX, Linux or Mac OS. Thus, a detailed evaluation is initiated by incorporating all types of OS. Moreover, a detection and response software tool could be implemented using the proposed technique which is left for future work. Research is expected to continue in this direction to further optimise the snapshot and transfer algorithms and the modules of the framework.

Further testing using higher-level complex scenarios and the fine integration with other existing cloud infrastructures can also help enhance the proposed solution. A new tool could be developed for the prediction of future worm attacks (Xiang *et al.*, 2009). Understanding how the current worms' infection propagate dynamically is important to safeguard against future work attacks. This can be done by investigating the trace left by the attacker which is considered as the attack pattern. Solutions for detection and prediction of worm propagation from one machine to another machine is needed. Thus, this study proposes a multi-step attack model for detecting and predicting the traditional worm by examining the various OSI layer's log from the worm source and the other machines are infected with it.

Based on this research, the cloud worm prevention system must be proposed to fight with the cloud worm (Vieira *et al.*, 2010). Firewalls and anti-virus programs try to block attacks, and IDS tries to identify attacks as they occur. Such techniques are critical for defence and the achievement of security but have limitations. A firewall can stop services by blocking certain port numbers but does little to evaluate traffic that uses allowed port numbers. IDS can evaluate traffic that passes through these open ports but cannot stop them. IPS can proactively block attacks. This research was conducted on windows environment; however, further research must be focused on

many other operating systems (i.e., Linux, Unix, Android, Mac OS and IOS) to secure future the cloud computing environment (Mohod & Alaspurkar, 2013).

The Proposed EGA can generate the characteristics of new types of cloud worm in each run. However, proving the efficiency of the algorithms is possible if it is tested on a real environment. However, managing a real testbed environment to find the accuracy percentage of future worm attacks is impossible. The testbed is where experiments can be done in a real cloud environment. In the testbed, the cloud environment is real, and the attack is real. The testbed can therefore be regarded as the production cloud. Creating a tool that can initiate new types of attack in that testing the accuracy of the proposed algorithm in terms of future attack detection is possible.

Moreover, given that one of the prospects of future work is the development of software tools, an effective approach to make the tool even more productive is desirable. Therefore, another future work plan is the application of the deep learning method for cloud worm detection and response. The proposed model achieved the highest percentage for correct classification, but deep learning helps in the use of these features to detect and respond to the cloud in the real world correctly and accurately. This helps the tool complete its action from pre-processing until the post-processing for worm detection and worm response. Applying the deep learning method within the context of security has the potential and is a promising field that can be explored in more detail in future research.