A Framework For Malware Detection Using Blacklist-Based Method

Saad Abdulhussein Abbas and Khairul Akram Zainol Ariffin
National University of Malaysia
Bangi, Selangor, Malaysia

***Abstract*—The security community needs to deal with an increasing number of malware samples that infect computer systems world-wide. The conventional approach is to assemble a blacklist of users that have been observed to be involved in malicious operations. The effectiveness of malware blacklists is poorly known. Examine in detail the completeness and accuracy of malware blacklists is the main challenges. The blacklists may become outdated if entries are not frequently updated by the providers. For this purpose, this study aims to address this issue by proposing a framework for the malware detection using a cloud blacklist-based approach. The proposed cloud blacklist will be updated by bringing new malware features from actual anti-malware namely as third parties. In order to do the matching between the current events and the blacklist's contents, three features have been used including hash value (i.e. MD5), fuzzy hash value (i.e. SSdeep) and size. The proposed method has been evaluated based on the common information retrieval metrics including precision, recall, and f-measure. Experimental results showed that the proposed method has the total ability to detect malware with precision of (75%), recall of (0.31) and f-measure of (0.441). This result is considered to be competitive compared to the state of the art blacklist-based approach.**

Keywords—Malware Detection, Signature-based, Blacklist, Hashing, Fuzzy Hashing, Cloud Malware Detection

I. INTRODUCTION

Malware is one of the threats that have been examined by many researchers. There are different terminologies for malware used by the computer science community such as malicious software, malicious code or malcode [1]. Malware is classified based on its behaviors such as viruses, worms and Trojan horses. Virus is a malicious code that replicates by inserting itself into host-based programs. While worm is a malicious code that replicates itself on the network layer [2]. Trojan horse is an embedded malicious code in a program or system to provide a backdoor for accessing the machine. The programs that are infected by Trojan are usually regular applications such as Microsoft Word and others [3].

For years, the research focuses on solving the problem with regard to the malware issue. Several approaches have been proposed to detect the malware in the wild. These approaches can be classified into two main groups including Signature-based techniques and Anomaly-based techniques [4]. Anomaly-based techniques aim to extract the features of executable programs [5]. In contrast, the signature-based techniques aim to extract features and characteristics of malware in order to analyse it and compare it with similar pattern [6].

Recently, there is a trend represented by taking the advantage of cloud malware detection. Cloud computing refers to any task such as storage, processing or even providing a platform that can be performed in the internet rather than locally [7]. Cloud would contribute toward enhancing the malware detection in different aspects. The main aspect can be represented by updating the blacklist of malware features with new features.

The security community needs to deal with an increasing number of malware samples that infect computer systems world-wide. The conventional approach is to assemble a blacklist of users that have been observed to be involved in malicious operations [8]. The effectiveness of malware blacklists is strongly considered in this study. Examine in detail the completeness and accuracy are the main challenges of malwares detection based on updating the blacklists. The blacklists may become outdated if entries are not frequently updated by the providers. That could be reducing the accuracy of detection the new malwares, which are not defined in the blacklists.

Nowadays, blacklists are strongly needed to be improved, when the non-public blacklists are sufficiently to against the variety of malware threats. The most blacklists are operated by antivirus (AV) vendors do not have effectively protect users from malwares, although integration an auto update blacklist would be straight-forward to improve the detection accuracy [9]. It's confidently noted that the auto update blacklists can help to improve the AV performance.

The specific difficulty is observing the user browsing activities where the user may vulnerable for different possible threats. Furthermore, blacklist of malware detection AV needs to be updated periodically in order to handle a new and unseen malware. The leakage of this process is acceptable for short time after last updating of the blacklist. Hence, this study will propose a framework for malware detection based on a cloud blacklist-based approach and using different features including size, hashing and fuzzy hashing.

II. RELATED WORK

Numerous methods have been proposed for the detection of malware for example, Gao et al. [10] have proposed an approach for detecting and characterizing social spam campaigns. The proposed method aims to detect spam over social networking specifically Facebook using a blacklist with an accuracy of 97%. However, the used blacklist needs to be updated.

Ma et al. [11] proposed a method to malicious web sites from suspicious URLs. The proposed method exploit the lexical features of a URL and compare it with a blacklist with an accuracy of 95%. Similarly, the used blacklist needs to be updated.

Sharifi & Siadati [12] proposes a new technique and architecture for a blacklist generator that maintains an up-to-date blacklist of phishing sites. The authors have addressed the problem of blacklist approach in which it needs to be updated periodically. Although the accuracy obtained was 91% however, the proposed method is limited regarding handling specific malware type which is phishing sites.

Ghafir & Prenosil [13] proposed a methodology for detecting any connection to or from malicious IP address. The authors have examined the problem of updating the blacklist to detect new malicious with an accuracy of 45%. However, accuracy still low and needs to be enhanced.

Zhou et al. [14] proposed a method for detecting Repackaged smartphone applications in third-party Android marketplaces. The authors have applied fuzzy hashing technique to effectively detect any changes occurred on the malware. Results showed that 13% of changed malware instances have been detected. However, accuracy still needs to be improved.

Agbefu et al. [15] proposed a Domain information based blacklisting method for the detection of malicious webpages. The authors have utilized an approximate matching method to leverage the changes that would occurred on the malware. However, the score of approximate similarity needs to be adjusted in order to improve the detection rate.

Walenstein et al. [16] proposed a method for comparing and searching binary programs to detect malware. The authors have addressed the huge numbers of minor variations of existing malware by proposing an approximate program matching with an accuracy of 100%. However, the dataset was relatively small.

Akiyama et al. [17] proposed a method for structural neighborhood of malicious URLs to improve blacklisting. Authors have focused on the problem that blacklists must be updated because malicious URLs tend to be short lived and their substrings may be partially mutated to avoid blacklisting. Results showed that 40% of unknown malicious URL. However, accuracy still need to be improved.

Prakash et al. [18] proposed a predictive blacklisting to detect phishing attacks using an approximate matching. The authors have addressed the problem of changes occurred on the malicious URL to avoid the detection. The proposed approximate matching algorithm leads to very few false positives (3%) and negatives (5%). However, threshold value of the approximate matching needs to be adjusted.

Rowe [19] proposed a contextual clues to malware using a large corpus. The proposed method aims to detect malware using specific features such as file name, size and hash value. Results showed that the accuracy was 70%. However, using modified versions of hash function such as fuzzy hash would.

Tao et al. [20] proposed a novel approach for classifying web pages automatically as either malicious or benign based on a supervised machine learning. Authors have examined different types of features that would discriminate the malicious web pages with an accuracy of 92.2%. However, the example set of malicious web pages was too limited.

Tseng et al. [21] proposed a new parallel automaton string matching approach. Authors studied the problem of matching the viruses and malicious web pages string patterns. Results showed an accuracy of 49.9%. Although the efficiency of the proposed method was substantial however, the accuracy of matching is still low.

Trapnell [22] Propose a multi-agent strategy for blacklisting malicious nodes in a peer-to-peer network. Authors have studied the problem of blacklisting malicious nodes in a network using an algorithm inspired by the immune system of the human body. The strategy can eliminate even a large, uniform distribution of malicious nodes in the network. However, the proposed method is sensitive to the parameters.

Meng & Kwok [23] have developed an adaptive blacklist-based packet filter using a statistic-based approach aiming to improve the performance of intrusion detection. The proposed method aims to improve the performance of detecting intrusions by blacklisting the signature of the intrusion. Results showed a significant enhancement regarding the time consumed toward detecting intrusion. However, signature matching in the blacklist is still consume long time.

Ziqian et al. [24] proposed a PageRank-improved algorithm that combined whitelist and blacklist for detecting malicious web pages. The proposed method was taking the advantage of both whitelist and blacklist with an accuracy of 82%. However, both whitelist and blacklist used in this study need to be updated.

Jang et al. [25] proposed a feature-rich hybrid anti-malware system, called Andro-Dumpsys, which leverages volatile memory acquisition for accurate malware detection and classification. Authors have addressed the similarity matching of malware creator-centric and malware-centric information with an accuracy of 99%. However, the proposed method has a long time consumption.

Ye et al. [26] have combined file content and file relations for cloud based malware detection. Authors have studied how file relations can be used to improve malware detection by combining file content and file relations together. The proposed method showed an accuracy of 99%. However, data used in the experiment was relatively small.

Meng [27] have proposed an adaptive non-critical alarm reduction using hash-based contextual signatures in intrusion detection. Authors have addressed the problem of lack of contextual information of intrusion by using hash function. Results showed an accuracy of 67.1%. However, the matching method cannot handle partial similarity such as prefixes and suffixes.

Meng & Kwok [28] presented a construction of contextual signatures with hash function in intrusion detection. Authors have addressed the problem of large amount of non-critical alarms will be generated during the detection process. The proposed scheme is compatible to different representations of intrusion detection signatures. However, another modification of hash function can be used to improve the accuracy such as the fuzzy hashing.

Chiba et al. [29] proposed a method for detecting malicious websites by learning IP address features. The detection has been performed using string-based features such

978-1-5386-0475-5/17/$31.00 ©2018 IEEE

as URL and DNS. However, results showed high rate of false positive (false alarms).

Kührer et al. [9] empirically analyze 15 public malware blacklists and 4 blacklists operated by antivirus (AV) vendors. Authors have examined different types of blacklist such as IP address blacklist and malicious URL blacklist. Results showed an accuracy of 58%. However, the features included in the utilized blacklist were too limited.

Das et al. [30] proposed a semantic-based online malware detection. Authors have examined the semantic features of a malware specifically n-gram which aims to address each token of words. Results showed an accuracy of 97%. However, different approaches can be used in further studies such as the hash function.

Chen et al. [31] proposed a static analysis for Android malware detection. Using a statistical approach, the executable files will compared based on the frequency of specific features which may indicate that file is a malware or not. Results showed an accuracy of 90%. However, more features can be used for the detection in further studies along with the statistical approach such as the hashing.

III. Research Framework

Basically, the proposed framework of cloud malware detection has been implemented using Visual Basic programming language and it is compatible with Windows operating system. The framework consists of three main components; cloud server, third-party and host as shown in Figure 1. Each aspect will be discussed in the following sub-sections.

A. Cloud Server

This component is considered to be the contribution of this study where a cloud server can be utilized to update the blacklist with new malware features. Such cloud server is a private server in which the end-user cannot access to it. This aspect will communicate with the third party aspect in order to get the new malware features via API 1. In addition, the cloud server will communicate with the host via API 2. API 2 is responsible to feed the host blacklist backup with the new information about the malware.

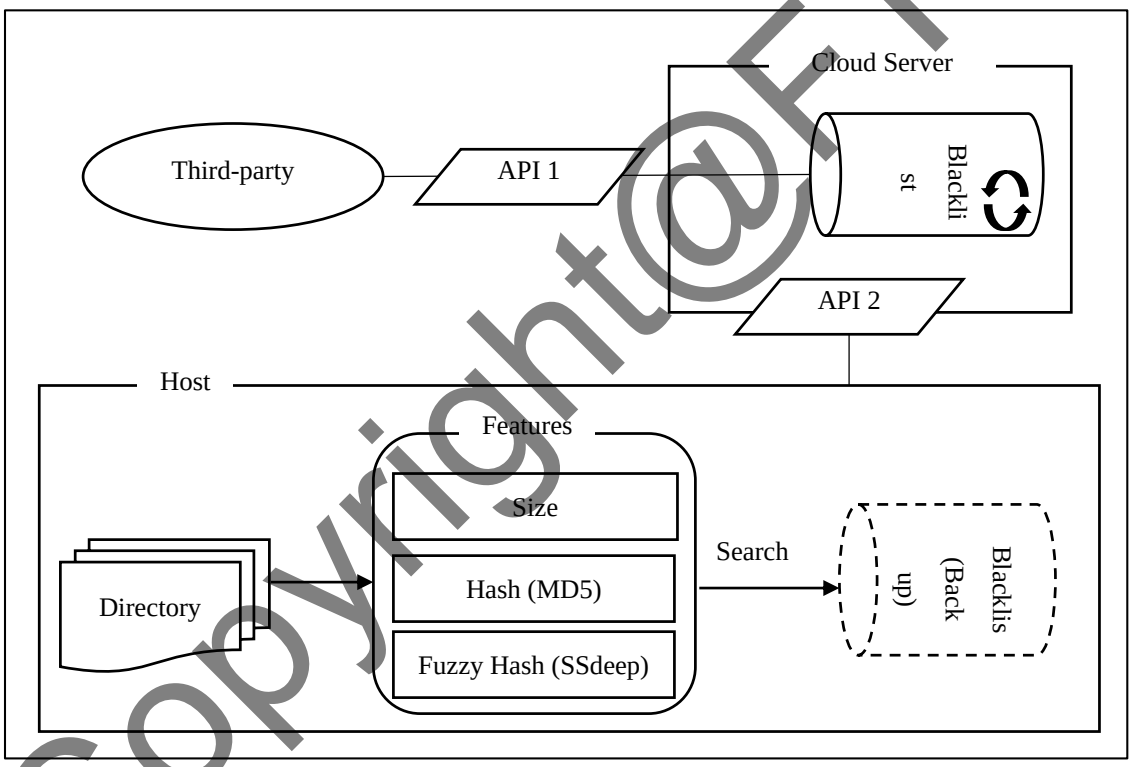

Figure 1 Framework of cloud malware detection

In fact, API 2 is considered to be an intermediate between the cloud server and the host. The communication between the API 2 and the host can be conducted via the HTTPS protocol where the host will make a request and the API 2 will respond with the new update of malware features as shown in Figure 2.

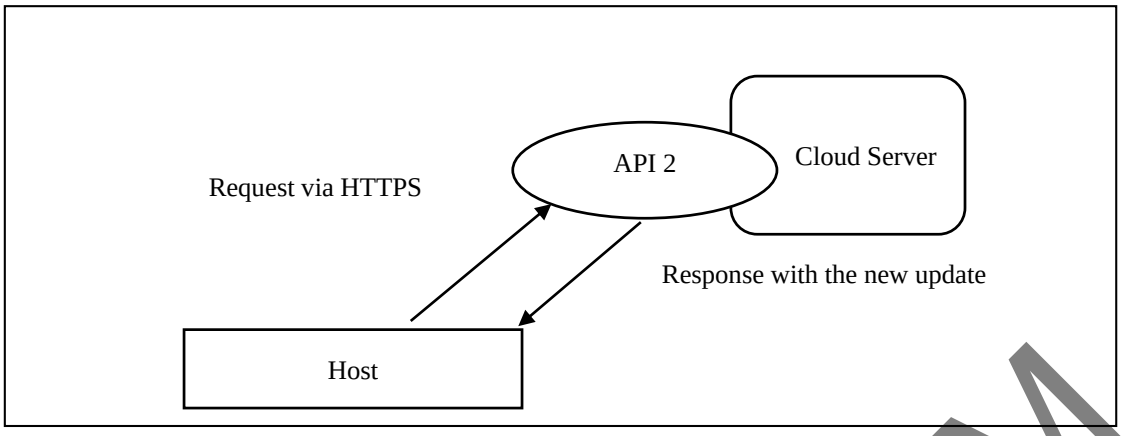

Figure 2 API and host communication

B. Third Party

This component is associated with an actual anti-malware that contains vast amount of information about the malware, it can be accessed via (https://virusshare.com). Such source contains about 3 million records of malware features including size, MD5 hashing, SHA-1 hashing, SHA-256 hashing and SSdeep fuzzy hashing. In addition, such anti-malware works by analysing real malware including viruses, Trojan horses and worms in order to extract signatures.

The update is conducting between the third party and the cloud server via an intermediator which is API 1. The update will be perform form 2 to 3 per minute automatically. The updated data will be structured in a relational manner. The mechanism of the update can be depicted in Figure 3.

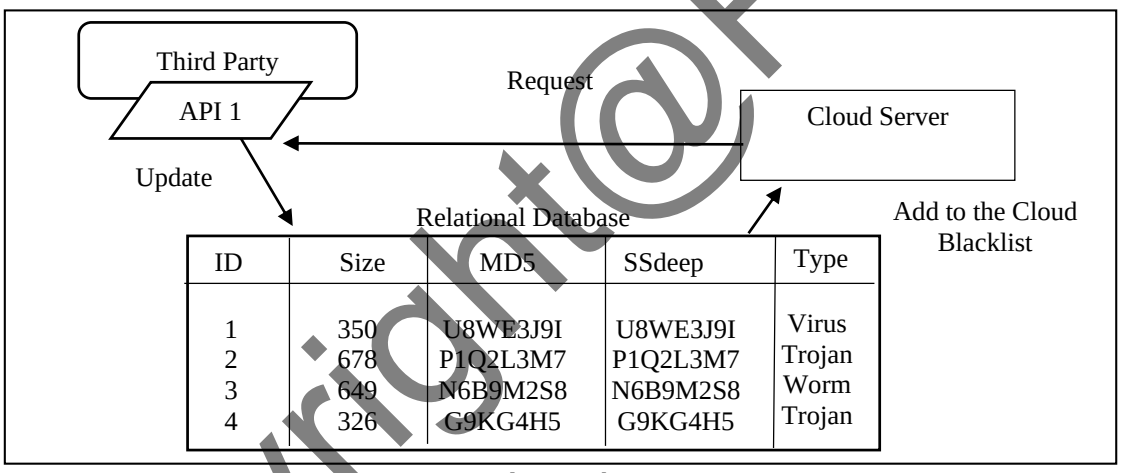

Figure 3 Update Mechanism

C. Host

This component is responsible for the client-side operation where the malware detection is being applied. First, this aspect can communicate with the cloud server through the HTTPS protocol in which a request is being asked to the API of the cloud server. Then, the new hash value of malware will be brought from the cloud server and stored in the host blacklist.

The hash values of the new malware are being brought in a JSON format. JSON is an open-standard file format that uses human-readable text to transmit data objects consisting of attribute. Once the system is being launched, the connection with the cloud server will be established. The aim behind the connection with the cloud server is to feed the host blacklist with the new malware features from the cloud blacklist. In this manner, if the connection is being lost, the host blacklist will be used as a backup for the malware detection process. Note that, the update from the cloud blacklist to the host blacklist is conducted every time the application is being launched.

Within the host, the process of malware detection will be applied by comparing the contents of the blacklist with the files in our partition in order to find any match. For this purpose, the hash value of the directory files will be acquired for the comparison with the hash values in the blacklist. To get the hash value of the directory files, Microsoft System Integrity Check has been used. Next section will discuss the detection in more details.

978-1-5386-0475-5/17/$31.00 ©2018 IEEE

D. Malware Detection

As mentioned earlier, our study will utilize a signature-based technique for the process of malware detection. Obviously, signature-based technique requires a predefined knowledge of the malware behaviour. Such knowledge could be a list or any information that guide the analysis of detecting the malware. In this research a blacklist of the malware features has been used. However, in order to map the actual processing programs with the predefined list, it is necessary to use a technique that has the ability to compare both information. In this manner, different features will be used including size, hashing and fuzzy hashing. Both blacklist and the features will be described in the following sub-sections.

1) *Blacklist*: This section aims to discuss the predefined knowledge used in this study in order to accommodate the malware detection. The blacklist used in this study has been brought from (https://virusshare.com). It contains different features of malware. As mentioned earlier, this blacklist will be updated with new instances once the application is being launched. Note that, the update has been performed automatically.
2) *Features*: In order to use the blacklist that has been illustrated in the previous sub-section, it is necessary to select specific features for the mapping. The mapping aims to compare the actual execution program features with the ones stored in the blacklist in order to detect the corresponding ones. To do so, this study utilizes the size, hash value (MD5) and fuzzy hashing (ssdeep) which can be defined as the mapping function that intended to compare data of arbitrary size with data of fixed size.

Since the information of the blacklist is in the form of hash values thus, it is necessary to get the hash value of the directory files in order to enable the comparison. In fact, the MD5 encryption has been used which has a length of 128 bit. The reason behind selecting such encryption lies on the efficient processing with its length..

In addition, the fuzzy hashing using SSdeep algorithm has been also used in order to address the approximate matching. This algorithm aims to examine the approximate matching between the malware's hash value and the blacklisting one. Unlike the complete matching, approximate matching aims to provide a percentage that indicates the similarity between the two files. Hence, instead of saying that a two files are either identical or not, it provides a value for indicating the similarity between them. In this manner, by using a threshold value, it is easy to decide whether a file is infected or not. For example, assume a two files with hash value as follows:

Hash value of File 1 = {EE49U50}
Hash value of File 2 = {EE49U31}

If the complete matching is being used, the results would refer that the two files are not identical. On the other hand, if the approximate matching (i.e. fuzzy hashing) is being used, the results can be provided as value of similarity such as 70%. Here, if we provide a threshold value of 70%, the two file can be matched. It can be interpreted as "if the similarity value = > 70% then the file is infected".

Now, it is possible to accommodate the comparison between the hash values of the directory files and the hash values in the blacklist.

IV. Results

This section aims to overview the results obtained by the proposed method based on the evaluation method described in the previous section. Table 4.4 shows the confusion matrix results. The following evaluation measures are used in the results:

• True positive (TP): the number of executables correctly classified as malicious.
• True negative (TN): the number of executables correctly classified as benign executables.
• False positive (FP): the number of executables mistakenly classified as malicious executables.
• False negative (FN): the number of malicious mistakenly classified as benign executables.
• Detection rate (DR): TP/(TP + FN).
• Accuracy (ACC): (TP + TN)/(TP + TN + FP + FN).

TP	TN	FN	FP
150	330	70	50

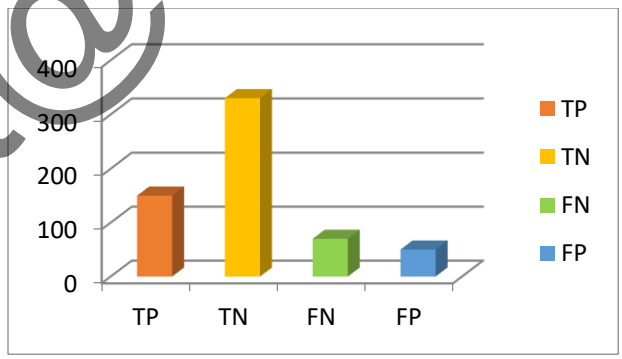

Table 1 Confusion matrix results for viruses

Category	TP	FP
Viruses	57	20
Worms	53	17
Trojan horses	40	13
Total	150	50

As shown in Table 1, the confusion matrix has been applied on the obtained results of the proposed method. First, the total number of each malware (i.e. viruses, worms and Trojans) has been declared, as well as, the true positive, false positive and true negative. Figure 4 depicts the results for each malware type.

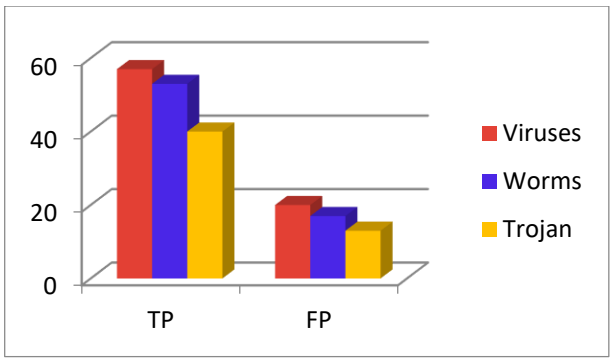

Figure 4 Results of TP and FP for each malware type

Based on the results of confusion matrix, Table 2 shows the results of precision, recall and f-measure.

Table 2 Results of precision, recall and f-measure

Category	Precision
Viruses	0.74025974
Worms	0.757142857
Trojan horses	0.754716981
Average	0.750706526

As shown in Table 2, each category has been examined in terms of precision, recall and f-measure. For instance, the viruses have obtained a precision of 0.74. Trojan horses have the precision 0.75. Rate of precision has been obtained by the worms where the precision was 0.75. Figure 5 depicts the results of precision value of 0.75, recall value of 0.31 and f-measure value of 0.441 for all malwares.

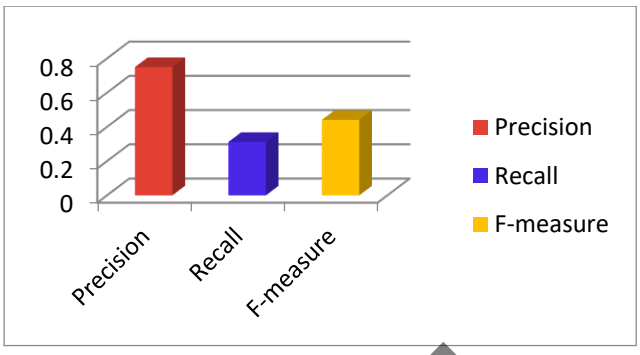

Figure 5 Results of precision, recall and f-measure for all malwares.

Basically, the proposed framework had the ability to detect infected files with low rate of false positive. This is demonstrating the effectiveness of the proposed method. However, Table 3 shows parameter setting of attaining the results.

Table 3 Parameter setting of the results

Parameter	Quantity
File scanned	600
Infected Files	200
Time elapsed	78 Sec.

The detection accuracy of the classifier is calculated by the equation:

$$Accuracy\ (\%) = \frac{TP + TN}{TM + TB} \qquad (1)$$

Where,
TP = True positive, the number of malwares correctly classified (150).
TN = True negative, the number of executables correctly classified (333).
TM = Total number of malwares (200).
TB = Total number of benign executables (400).

It is find that all the classifier overall performed with correctness of more than 80 % of detection accuracy.

Table 4. Parameter Confusion matrix results

Measure	Value	Derivations
Sensitivity	0.6818	TPR = TP / (TP + FN)
Specificity	0.8684	SPC = TN / (FP + TN)
Precision	0. 750	PPV = TP / (TP + FP)
Negative Predictive Value	0. 825	NPV = TN / (TN + FN)
False Positive Rate	0. 1316	FPR = FP / (FP + TN)
False Discovery Rate	0.2500	FDR = FP / (FP + TP)
False Negative Rate	0. 3182	FNR = FN / (FN + TP)
Accuracy	0. 8000	ACC = (TP + TN) / (P + N)
F1 Score	0. 7143	F1 = 2TP / (2TP + FP + FN)
Matthews Correlation Coefficient	0. 5625	TP*TN - FP*FN / sqrt((TP+FP)*(TP+FN)*(TN+FP)*(TN+FN))

V. Discussion

In order to determine the effectiveness of the results obtained by the proposed method, it is necessary to accommodate a comparison with the state of the art blacklist approaches. However, the evaluation criteria that have been used to validate a malware detection method were various. Some studies have used the time consumption as an evaluation measure, while others have addressed the precision of detection. Even the studies that have focused on the precision showed different metrics for the validation where some of them have used precision only and other used number of correct infected and incorrect infected. For this purpose, the comparison in this section will be based on the True Positives and False Positives. After that, it is easy to compute the precision. Table 5 shows the results of the related work against the proposed method.

Table 5. Comparisons with state of the art blacklist approaches

Author	Precision	TP	FP	Limitation
[9]	91%	-	11%	Limited features included in the utilized blacklist and need to be update
[32]	70%	20%	10%	An attacker might try to employ recently infected hosts whose IP addresses have not been blacklisted yet.
[25]	99%	14%	45%	Long time consumption
Our proposed framework	75.%	68%	13 %	More features want to be considered

As shown in Table 5, the results of precision for the study Coskun [32]

Was (70%). Whereas, our proposed method precision result was (75%). It is obvious that our proposed method has outperformed the one proposed by Coskun [32].

However, the study of [9] has obtained a precision of 0.91% which slightly better than our proposed method.

Generally, our proposed method has showed competitive performance in which the precision of detecting malware using precision was better than some state of the art and relatively close to others. The proposed approach is always required to connect to the internet, which can be demonstrated as usefulness of the proposed cloud blacklist approach.p5

vI. Conclusion

This study proposed a framework for malware detection based on a cloud blacklist-based approach. Such cloud blacklist aims to update its contents automatically in order to detect new malware. In addition, three types of features have been used including size, MD5 and fuzzy hash such ssdeep to map/similar correspondences between scanning files and blacklist instances. The proposed method has been evaluated based on the common information retrieval metrics including precision, recall, f-measure. Experimental results showed that the proposed method has the ability to detect malware with precision of (75%). Such results reflect the usefulness of using the proposed cloud blacklist.

In the future researches, examining the dynamic features would yield to improve the accuracy of detection.

REFERENCES

[1] K. Mathur and S. Hiranwal, "A survey on techniques in detection and analyzing malware executables," *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 3, 2013.

[2] P. Mateti, "Viruses, Worms and Trojans," ed, 2006.

[3] L. A. Hughes and G. J. DeLone, "Viruses, worms, and trojan horses: Serious crimes, nuisance, or both?," *Social science computer review*, vol. 25, pp. 78-98, 2007.

[4] M. Siddiqui, M. C. Wang, and J. Lee, "A survey of data mining techniques for malware detection using file features," in *Proceedings of the 46th annual southeast regional conference on xx*, 2008, pp. 509-510.doi.

[5] A. Patcha and J.-M. Park, "An overview of anomaly detection techniques: Existing solutions and latest technological trends," *Computer networks*, vol. 51, pp. 3448-3470, 2007.

[6] A. Amamra, C. Talhi, and J.-M. Robert, "Smartphone malware detection: From a survey towards taxonomy," in *Malicious and Unwanted Software (MALWARE), 2012 7th International Conference on*, 2012, pp. 79-86.doi.

[7] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, and I. Stoica, "A view of cloud computing," *Communications of the ACM*, vol. 53, pp. 50-58, 2010.

[8] C. Rossow, C. J. Dietrich, C. Grier, C. Kreibich, V. Paxson, N. Pohlmann, H. Bos, and M. Van Steen, "Prudent practices for designing malware experiments: Status quo and outlook," in *Security and Privacy (SP), 2012 IEEE Symposium on*, 2012, pp. 65-79.doi.

[9] M. Kührer, C. Rossow, and T. Holz, "Paint it black: Evaluating the effectiveness of malware blacklists," in *International Workshop on Recent Advances in Intrusion Detection*, 2014, pp. 1-21.doi.

[10] H. Gao, J. Hu, C. Wilson, Z. Li, Y. Chen, and B. Y. Zhao, "Detecting and characterizing social spam campaigns," in *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement*, 2010, pp. 35-47.doi.

[11] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker, "Beyond blacklists: learning to detect malicious web sites from suspicious URLs," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2009, pp. 1245-1254.doi.

[12] M. Sharifi and S. H. Siadati, "A phishing sites blacklist generator," in *Computer Systems and Applications, 2008. AICCSA 2008. IEEE/ACS International Conference on*, 2008, pp. 840-843.doi.

[13] I. Ghafir and V. Prenosil, "Blacklist-based malicious IP traffic detection," in *2015 Global Conference on Communication Technologies (GCCT)*, 2015, pp. 229-233.doi:10.1109/GCCT.2015.7342657.

[14] W. Zhou, Y. Zhou, X. Jiang, and P. Ning, "Detecting repackaged smartphone applications in third-party android marketplaces," in *Proceedings of the second ACM conference on Data and Application Security and Privacy*, 2012, pp. 317-326.doi.
[15] R. E. Agbefu, Y. Hori, and K. Sakurai, "Domain information based blacklisting method for the detection of malicious webpages," *International Journal of Cyber-Security and Digital Forensics (IJCSDF)*, vol. 2, pp. 36-47, 2013.
[16] A. Walenstein, M. Venable, M. Hayes, C. Thompson, and A. Lakhotia, "Exploiting similarity between variants to defeat malware," in *Proc. BlackHat DC Conf*, 2007.
[17] M. Akiyama, T. Yagi, and M. Itoh, "Searching structural neighborhood of malicious urls to improve blacklisting," in *Applications and the Internet (SAINT), 2011 IEEE/IPSJ 11th International Symposium on*, 2011, pp. 1-10.doi.
[18] P. Prakash, M. Kumar, R. R. Kompella, and M. Gupta, "Phishnet: predictive blacklisting to detect phishing attacks," in *INFOCOM, 2010 Proceedings IEEE*, 2010, pp. 1-5.doi.
[19] N. C. Rowe, "Finding contextual clues to malware using a large corpus," in *Computers and Communication (ISCC), 2015 IEEE Symposium on*, 2015, pp. 229-236.doi.
[20] W. Tao, Y. Shunzheng, and X. Bailin, "A novel framework for learning to detect malicious web pages," in *Information Technology and Applications (IFITA), 2010 International Forum on*, 2010, pp. 353-357.doi.
[21] K.-K. Tseng, Y.-D. Lin, T.-H. Lee, and Y.-C. Lai, "A parallel automaton string matching with pre-hashing and root-indexing techniques for content filtering coprocessor," in *Application-Specific Systems, Architecture Processors, 2005. ASAP 2005. 16th IEEE International Conference on*, 2005, pp. 113-118.doi.
[22] B. C. Trapnell, "A peer-to-peer blacklisting strategy inspired by leukocyte-endothelium interaction," in *International Conference on Artificial Immune Systems*, 2005, pp. 339-352.doi.
[23] Y. Meng and L.-F. Kwok, "Adaptive blacklist-based packet filter with a statistic-based approach in network intrusion detection," *Journal of Network and Computer Applications*, vol. 39, pp. 83-92, 2014.
[24] Y. Ziqian, Z. Wenhui, F. Huijuan, and T. Zhixiao, "A PageRank-improved ranking algorithm based on cheating similarity and cheating relevance," in *Computer and Information Science (ICIS), 2017 IEEE/ACIS 16th International Conference on*, 2017, pp. 257-263.doi.
[25] J.-w. Jang, H. Kang, J. Woo, A. Mohaisen, and H. K. Kim, "Andro-dumpsys: anti-malware system based on the similarity of malware creator and malware centric information," *computers & security*, vol. 58, pp. 125-138, 2016.
[26] Y. Ye, T. Li, S. Zhu, W. Zhuang, E. Tas, U. Gupta, and M. Abdulhayoglu, "Combining file content and file relations for cloud based malware detection," in *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2011, pp. 222-230.doi.
[27] Y. Meng, "Adaptive non-critical alarm reduction using hash-based contextual signatures in intrusion detection," *Computer Communications*, vol. 38, pp. 50-59, 2014.
[28] Y. Meng and L.-f. Kwok, "A generic scheme for the construction of contextual signatures with hash function in intrusion detection," in *Computational Intelligence and Security (CIS), 2011 Seventh International Conference on*, 2011, pp. 978-982.doi.
[29] D. Chiba, K. Tobe, T. Mori, and S. Goto, "Detecting malicious websites by learning IP address features," in *Applications and the Internet (SAINT), 2012 IEEE/IPSJ 12th International Symposium on*, 2012, pp. 29-39.doi.
[30] S. Das, Y. Liu, W. Zhang, and M. Chandramohan, "Semantics-based online malware detection: towards efficient real-time protection against malware," *IEEE transactions on information forensics and security*, vol. 11, pp. 289-302, 2016.
[31] P. Chen, Z. Rong-Cai, S. ZHENG, X. Jia, and Y. Li-Jing, "Android Malware of Static Analysis Technology Based on Data Mining," *DEStech Transactions on Computer Science and Engineering*, 2016.
[32] B. Coskun, "(Un) wisdom of Crowds: Accurately Spotting Malicious IP Clusters Using Not-So-Accurate IP Blacklists," *IEEE Transactions on Information Forensics and Security*, vol. 12, pp. 1406-1417, 2017.