*International Journal of Engineering & Technology*, 7 (4.15) (2018) 120-125

International Journal of Engineering & Technology

*Website: www.sciencepubco.com/index.php/IJET*

***Research paper***

# Signature-Based Malware Detection Using Sequences of N-grams

**Alogba Moshood Abiola[1], Mohd Fadzli Marhusin[2]***

*[1]Faculty of Science and Technology, Universiti Sains Islam Malaysia (USIM), 71800 Nilai, Negeri Sembilan, Malaysia*
*[2]Cybersecurity and Systems Research Unit, Islamic Science Institute (ISI), Universiti Sains Islam Malaysia (USIM), 71800 Nilai, Negeri Sembilan, Malaysia*
**Corresponding author E-mail: fadzli@usim.edu.my*

**Abstract**

The focus of our study is on one set of malware family known as Brontok worms. These worms have long been a huge burden to most Windows-based user platforms. A prototype of the antivirus was able to scan files and accurately detect any traces of the Brontok malware signatures in the scanned files. In this study, we developed a detection model by extracting the signatures of the Brontok worms and used an n-gram technique to break down the signatures. This process makes the task to remove redundancies between the signatures of the different types of Brontok malware easier. Hence, it was used in this study to accurately differentiate between the signatures of both malicious and normal files. During the experiment, we have successfully detected the presence of Brontok worms while correctly identifying the benign ones. The techniques employed in the experiment provided some insight on creating a good signature-based detector, which could be used to create a more credible solution that eliminates any threats of old malware that may resurface in the future.

***Keywords***: *N-grams; K-grams; Signature-based detection.*

## 1. Introduction

PC malware has been detected for a little over 30 years with the nature of the threats having changed in a very surprising way in the last two decades. The threats in today's systems are extremely complicated. A lot of today's malware includes significant levels of code of different variety and instructions. The codes include a wide range of array of trojans, exploits, rootkits, phishing scams and spywares, as well as viruses and worms that are purposely built to take over a user's machine for either extorting the user or for some other harmful reasons. In particular, in the new age of globalisation, in which there is extensive use of technologies and connectivity provided by the Internet, launching attacks on victims can be too quick and easily accomplished [1].

Furthermore, malware codes can be embedded in e-mails, bogus software packages or trojans on a website that hosts downloads (movies, etc.) where they are waiting to be mistakenly or purposely clicked and then get automatically installed on the victim's machine without a user's knowledge. Malware had increased significantly in recent years without any stoppage. According to the Kaspersky Lab AV, in 2009, the databases contained more than 570,500 records and around 3,500 new records were added on a weekly basis. Subsequently, in 2015, the Kaspersky Lab Solutions detected and repelled a total of 235,415,870 malicious attacks from online resources located all over the world with most of them originating from other sectors [2].

A signature-based AV system is another form of malware detection program that is designed in such a way that the malware signatures are stored in its database. The anti-malware itself, then alerts the computer, on which it is installed. Any malware signature that matches the database, will be detected on the system. Basically, the signature-based AV scans the system's files, user files and unused locations to search for any signs of the malware's signature, which is stored in its database. In a case of a malware signature is found, the AV alerts and performs pre-set tasks by cleaning up, moving them into quarantine or deleting the files. Signature-based AV can stop any known malware and is considered effective for detecting and removing a malware that has already infected a computer system.

In this study, a new signature-based AV was developed. For experiments and evaluation, a Brontok family that worked against normal exe files on a Windows-based platform was used to measure the detection accuracy of the AV that was to be developed.

## 2. Background of Study

In this section, we will provide information and explain the technical details of the malware used in this study.

### 2.1. Brontok Family

Brontok is a worm originated from Indonesia. This worm has a specific intent of sending political messages against pollution and immorality. These messages were used to attack the pleasure seekers and the Israeli government before it became a worldwide threat to others. When a Brontok malware runs on a computer, it makes a copy of itself within the user's application directory and it sets itself up to start with the Windows operating system. Moreover, it has the capability to disable the regedit.exe application and the register editor in Windows, and turns off the computer's firewall [3].

A Brontok malware is a family of mass-mailing email worms [4], which can spread out by sharing copies of themselves as attachments via emails that are sent to the addresses that have been iden-

tified on the infected computer. Moreover, they may copy themselves onto a USB or a pen drive. They can disable an AV and immediately terminate certain applications that might be a threat to them and, if any changes are made, they force Windows to immediately restart [5]. In this manner, the family is quite widespread. Some of the names given to the malware are *W32/Rontokbro.gen@MM, W32.Rontokbro@mm, Back-Door.Generic.1138, W32/Korbo-B, Worm/Brontok.a, Win32.Brontok.A@mm, opopopopo, about.Brontok.a, Worm.Mytob.GH, W32.Brontok.a, W32/Brontok.C.worm, Win32/Brontok.E,W32.Rontokbro.D@mm* and *IWorm.VB.DV*.

### 2.2. Characteristics of Brontok Malware

The Brontok malware is a very complicated and annoying worm, affecting Windows users all over the world. Figure 1 depicts a screen capture of an affected PC.


**Fig. 1:** PC affected by Brontok malware

There are several characteristics of the Brontok malware. They have many variants and come under different names, for example, *Brontok.A, Brontok.C, Y2K, Milenium2K, You and Me, File Eater*, and etc. Their variants can damage important files on the hard drive. On download and execution, they secure their present in the infected PC by altering the registry and file structures. Moreover, on any attempt to delete the malware in directories, they will be immediately restored [5]. During its heyday, many AVs could not detect these malwares until their signature was added to the database. On successful penetration, they clone themselves into most of the directories and persist. In fact, they even proactively deny any users who attempt to search for online information about their removal. In addition, they can deny the access to certain parts of the operating system.

## 3. Technical Details

This section provides a brief description of a Brontok malware family when it infects a computer system. For this purpose, W32/Brontok.N is considered as an example [5]. W32/Brontok.N is a well-known complex email worm that can disable any AV software crossing its path. It can clone itself on the local hard disk of a Windows operating system and ensures that its eradication will be a difficult task for the victim. This malware, W32/Brontok.N, was identified at the end of March in 2006 [6].

### 3.1. Installation

When the worm's file starts, it duplicates itself with several aliases within several folders located on the local hard drive. The file names in some cases may be any of the following: csrss.exe, inetinfo.exe, lsass.exe, services.exe, smss.exe, or winlogon.exe.
Some of the worm's files exhibit a hidden system and read-only characteristics. In addition, it can develop files with COM, EXE and PIF extensions. Note that this Brontok worm produces multiple set-up points, such as the start-up registry keys and scheduled jobs, for creating duplicated files. Consider the following example:

Status ID Day Time Command
1 Each M T W Th F S Su 5:08 PM "C:\Documents and Settings\User\Local Settings\Application Data\jalak-931976415-bali.com"
2 Each M T W Th F S Su 11:03 AM "C:\Documents and Settings\User\Local Settings\Application Data\jalak-931976415-bali.com"
The Brontok worm manifests few text files on the victim's local hard disk with the name baca bro!!!.txt, which is placed at the root of the system. Here is a text example of the instruction:
BRONTOK.C
Sedikit Jawaban u/ Membungkam Mulut Sesumbar 'MEREKA'.
Nobron = Satria Dungu = Nothing !!! Romdil = Tukang Jiplak = Nothing !!!
Nobron & Romdil -->>
Kicked by The Amazing Brontok
[ By JowoBot ]
The c.bron.tok.txt file holds the following types of text: (i) Brontok.C and (ii) By:JowoBot. It keeps several duplicates of itself in the memory. In a propagation example, the worm sends out its messages with the following contents: Fotoku yg Paling Cantik and My Best Photo. The body of the message can be any of the following:
Hi, Aku lg iseng aja pengen kirim foto ke kamu. Jangan lupain aku ya !. Thanks, Hi, I want to share my photo with you. Wishing you all the best. Regards,
The name of the attachment is photo.zip. To gather the e-mail addresses for further distribution, the worm searches the drives from C: to Z: for any available e-mail address in files having the following extensions: .asp, .bat, .cfm, .com, .com, .csv, .doc, .eml, .exe, .htm, .html, .php, .pif, .ppt, .scr, .txt, .wab, and .xls. Then, the detected addresses are verified against a large item of strings. If in any situation, some part of the e-mail address that is detected corresponds to the content in that item, then such e-mail addresses is dumped or discarded. Then, the worm develops itself in folders with names like Spread.Mail.Bro and Spread.Sent.Bro, which are created as hidden subfolders on the system. The folder that comes first has the list of e-mail addresses that were gathered from the infected computer.
Each e-mail address is expressed by the file with the same text as that of the c.bron.tok.txt file. When an email is sent to an e-mail address by the worm, a similar file is then moved to the second folder to avoid sending the file to the same address again.

## 4. Methodology

In this section, we will explain the methodology used in this study, which include the process model, the dataset, dataset transformation and preparation for the detection purpose.

### 4.1. Process Model

The process model is shown in Figure 2. We begin our research by identifying the problem and defining the aim and need for such research. Our intent is to have a better capability for signature-based malware detection; hence, this study is necessary. Then, using the data gathered, we proceed with the requirement phase, which involves collecting Brontok malware file samples that are required to test the AV.
In the next phase, i.e. the design and development, a code is required for all actions that would have to be performed by the AV (such as the scan, detection and removal of any form of the malware) and a well-designed interface would have to be developed.

For demonstration and evaluation, an accuracy test for detection was performed.

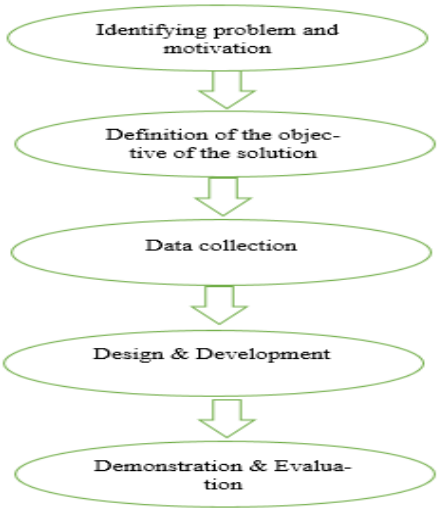


**Fig. 2:** Design science research process model.

Figure 3 depicts the flowchart of the detection process. Generally, this process involves collecting a dataset from reputable sources. The signatures of the datasets are labelled as B and M (for malware); moreover, those with the label M are extracted. We then created two dedicated repositories for both B and M. The redundant signatures coexisting in both B and M are removed. Subsequently, a refined version of B and M is used as a signature database for detection. A test sample is then evaluated against the signature database and the results are reported.

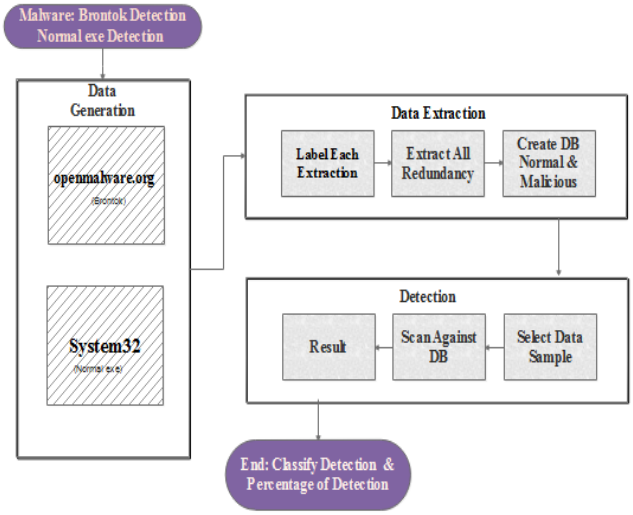


**Fig. 3:** Flowchart of the detection process.

## 5. Data Collection

For this study, a Brontok malware family's dataset was established. The malware files were downloaded from online sources such as VX Heaven [6] and Open Malware [7], and then stored on a virtual machine. Some of the files were collected from trusted websites via Google search, or from sites having collections of malware such as VirusSign and VirusShare.

### 5.1. N-Gram

There are several ways for evaluating the effectiveness of any AVs. One way is by examining the ratio of good detection rate of the malware, e.g. The Brontok malware family. However, the false positive ratio that is generated should be considered. A high false positive or false negative would render the AV unusable; hence, a thorough understanding of the usage of n-grams technique, along with any of the efficient classification algorithms, would help for a satisfactory detection [8].

The development of n-grams requires breaking down a large string of data into several substrings with a fixed n length. The nature of the dataset that is used for establishing the knowledge base and testing can be enormously large. Nevertheless, the only way to understand the pattern in the data is by breaking it into multiple parts. The size of the dataset can then be reduced by removing redundancies within this new dataset. Therefore, it is in our best interest to try and select a small amount of relevant and useful substrings from the large strings of the data, which would help achieve an efficient accuracy for the classification. For example, the string 'FEELFREE', when broken down to a substring of n=4 grams, leads to terms like "FEEL", "ELFR", "LFRE", "FREE" and so on. Additional details on preparing n-grams can be found in [9].

### 5.2. Dataset Preparation and Design

To determine whether a malware is genuinely malicious in nature or not, verifying the authenticity of the malware dataset is important. The simplest and quickest way is utilising a free online service such as VirusTotal [10], which analyses any type of files and identifies if they contain a malicious payload in their signature. A sample outcome obtained when scanning such file is shown in Figure 4-6.

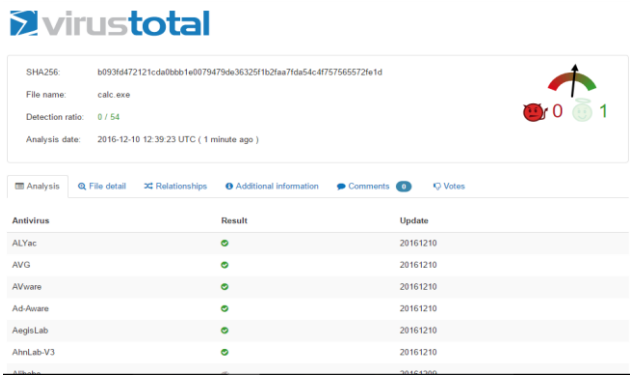


**Fig. 4:** Outcome of calc.exe, a calculator program.

Note that the normal dataset used in this experiment was executable files that can be found in any Windows-based computer system within the system directory, i.e.; C:\Windows\System32. This is the easiest way of getting a trusted benign dataset as these files are widely available across all Windows OS-based PCs. Having a benign dataset is important as it helps to ensure that the detector works by distinguishing both malware files and benign ones. Thus, without having a benign dataset, all executable in the dataset could simply be flagged as malwares, which would lead to a 100% true positive result; however, such result would be misleading.

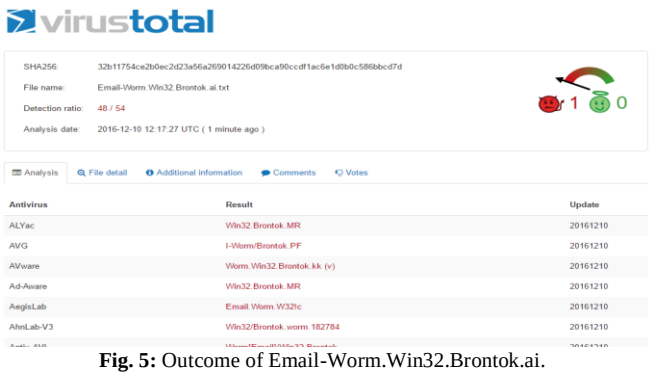


**Fig. 5:** Outcome of Email-Worm.Win32.Brontok.ai.

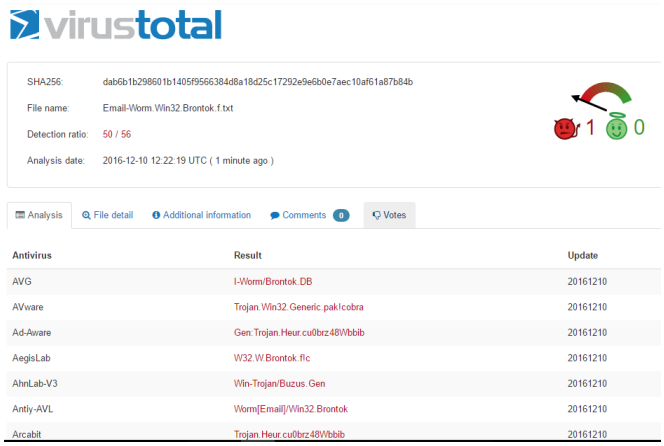


**Fig. 6:** Outcome of Email-Worm.Win32.Brontok.f.

The malicious files used in this study are derived from the Brontok Malware family. Labelling malware with their proper malware family assists in their understanding and tracking. 24 raw Brontok Worm samples were collected from openmalware.org [7] using a link from the Zeltser Security Corp [11]. Each downloaded Brontok file was labelled with its original filename, and the date and other information on the malware was added to the site. In order to allow accessing the Brontok file without accidentally executing it, it was necessary to change the extension from .exe to .txt file. However, that does not mean it is no longer executable and dangerous.

## 5.3. Extraction Process

Both normal and malicious files were acquired. We used Hex Editor (HxD) to view the content of the files. The data shown in the viewer confirmed that those files were legitimate. We use this view data and the MD5 hash values of the files to determine the authenticity of the files. Some signatures might have been altered, but the viewer was able to show every file's true identity. Figure 7 and 8 depict two examples of benign and malign files. Both files have some signatures in common because they are both .exe files.
The dataset which contains benign and malicious files was stored in two distinct directories. Our C# program grabbed the binary stream of these files and then stored each in temporarily files, acting as intermediate files. Every single executable's stream of binary information had to be regarded as a string of bytes, and was first extracted and transformed into base-64 bytes. Then, the strings were tokenised into n-grams of 5 characters per item as shown in Figures 9 and 10. The length was in the range of 100 to 400000. Using these techniques for the extraction and data transformation provided us with a clear understanding of the whole detection process.

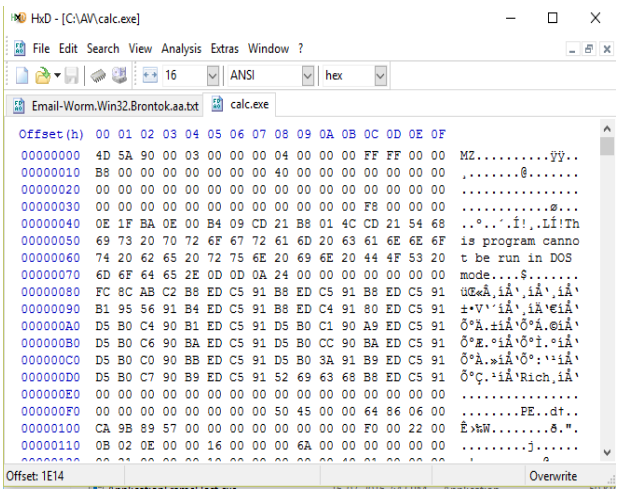


**Fig. 7:** Hex list of a normal calc.exe header.

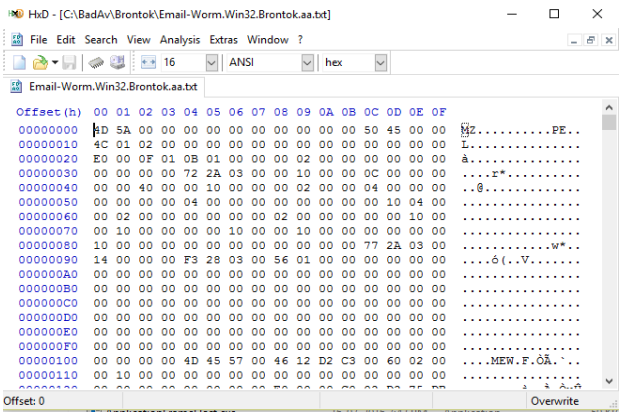


**Fig. 8:** Hex list of malicious Brontok.aa.exe header.

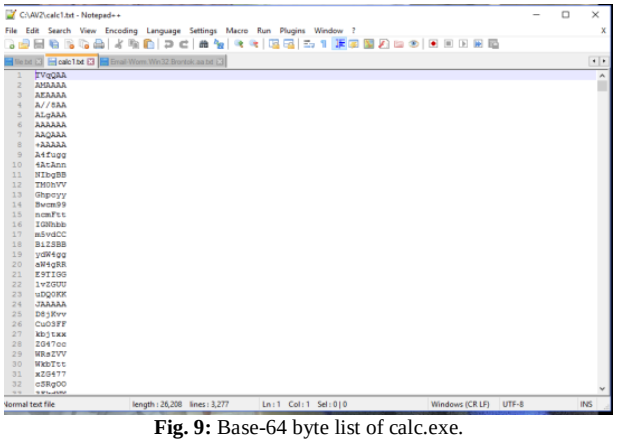

**Fig. 9:** Base-64 byte list of calc.exe.

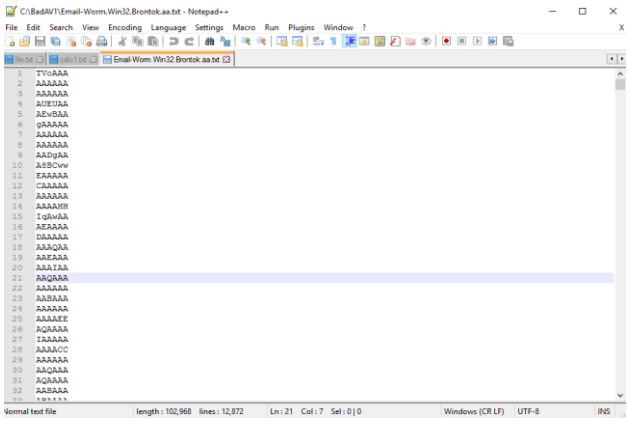

**Fig. 10:** Base-64 byte list of malicious Brontok.aa.exe.

Table 1 depicts the branches of Brontok family, which were selected for this study.

**Table 1:** Malware type (Brontok Family)

| Raw Sample | File Size (KB) | No of n-grams |
|---|---|---|
| Email-Worm.Win32.Brontok.aa | 100 | 102,968 |
| Email-Worm.Win32.Brontok.ai | 380 | 389,936 |
| Email-Worm.Win32.Brontok.bz | 94 | 97,216 |
| Email-Worm.Win32.Brontok.cd | 88.9 | 91.064 |
| Email-Worm.Win32.Brontok.de | 214 | 219,832 |
| Email-Worm.Win32.Brontok.dl | 214 | 219,832 |
| Email-Worm.Win32.Brontok.dq | 204 | 209,712 |
| Email-Worm.Win32.Brontok.e | 170 | 174,760 |
| Email-Worm.Win32.Brontok.f | 80 | 81,920 |
| Email-Worm.Win32.Brontok.n | 89.7 | 91,888 |
| Email-Worm.Win32.Brontok.o | 128 | 131,736 |
| Email-Worm.Win32.Brontok.q.exe | 88.9 | 91,064 |
| Email-Worm.Win32.Brontok.s | 231 | 237,024 |
| Email-Worm.Win32.Brontok.u | 128 | 131,072 |
| W32 Rontokbro.gen@MM.zip | 68.1 | 69,784 |

| | | |
|---|---|---|
| 5bc7b92869dc48118c4c43ed2a657481.exe Win32.Rontokbro.H | 259 | 265,416 |
| 25f32b681f575afa5133a7e39cc2ee4d.exe Win32.Rontokbro.H | 245 | 251,224 |
| 43EC3222F89CE913BDC295528261B6E8.out Worm.Brontok.I | 94.8 | 97,088 |
| 4514db26ab3c23c91b0ec7c8b6cc1d24 WIN.Worm.Brontok | 474 | 486,048 |
| 6029075.eeea1a8fe37750a133fc42387e03b36d WIN.Worm.Brontok | 771 | 789,920 |
| 6299765.8cfd65cbae03c471fb9705d1db36f8e0 Worm.Brontok-8 | 276 | 282,896 |
| 8375260.859cb9207ba1bb74958e8beae7598a7d WIN.Worm.Brontok | 551 | 564,696 |
| c8739c743fcd3008c6cc80878a962873.exe Win32.Worm.Brontok.BO | 89.5 | 91,744 |
| db36f443505ed4cad8365edfc4bb134b WIN.Worm.Brontok | 264 | 270,840 |
| (Average: 213,699, Max: 789920, Min: 69784) | | |

Table 2 contains a list of several benign executables derived from Windows with their file sizes in KB, and the number of base-64 converted strings of n-grams.

**Table 2:** Benign Type (Normal EXE)

| Sample | File Size (KB) | No of n-grams |
|---|---|---|
| Acu.exe | 48 | 49,152 |
| Alg. Exe | 199 | 204,256 |
| Appidtel.exe | 57.5 | 58,976 |
| AtBroker.exe | 116 | 119,056 |
| Audit.exe | 166 | 170,392 |
| Baaupdate.exe | 234 | 240,296 |
| Bcdboot.exe | 360 | 369,184 |
| BioIso.exe | 450 | 461,720 |
| BitLockerWizard.exe | 213 | 218,456 |
| Bootcfg.exe | 185 | 190,048 |
| Bootsect.exe | 221 | 226,848 |
| browser_broker.exe | 55.1 | 54,456 |
| Bthudtask.exe | 84.2 | 86,288 |
| ByteCodeGenerator.exe | 106 | 109,224 |
| Calc.exe | 67.1 | 68.808 |
| CameraSettingsUIHost.exe | 65.6 | 67,254 |
| CastSrv.exe | 150 | 154,312 |
| regedt32.exe | 24.5 | 25,120 |
| rundll32.exe | 145 | 148,546 |
| write.exe | 23.4 | 24,032 |
| (Average: 152,421, Max: 461720, Min: 24032) | | |

Table 1 and 2 contain the original extraction data before the redundancies in the collection of n-grams were further refined. When comparing the average size of the generated n-grams, we found that the average size of the malware n-grams was significantly greater than the benign files. The detection program we developed and used was written in C#.NET as illustrated in Figure 11.

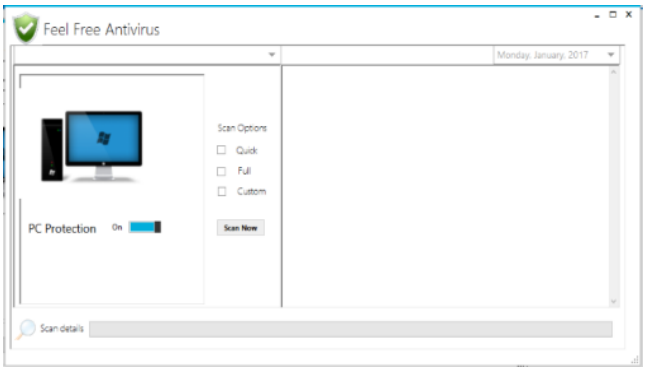


**Fig. 11:** Platform of the AV.

### 5.4. Performance Measures

We evaluate the performances of the detector using one or more of the following:

- TP: The percentage of malware executables correctly classified as malware;
- FP: The percentage of benign programs wrongly classified as malware;
- TN: The percentage of benign executables correctly classified as benign;
- FN: The percentage of malware programs wrongly classified as benign;
- Accuracy: The ratio of correctly classified malware and benign executables against the outcome of the entire set.

## 6. Results and Discussion

To determine whether the outcome of this experiment satisfies the objectives, we revisit the objectives of the research, which include:

- To study the general architecture of Brontok family.
- To study how the Brontok family infiltrates a computer system.
- To recognise efficient ways to detect Brontok.
- To measure the performance and accuracy of Brontok detection.

To improve the quality of the knowledge base of benign B and malware M signatures, n-grams in B which also contain a duplicate in M or vice versa were removed. There was a possibility that removing the duplicates would cause false detection. The size of the n-gram could also influence the accuracy. Moreover, it was possible that n = 5 was not an optimal value, causing too many duplicates between B vs. M with a subsequent risk of false detection.

Eventually, the final versions of B and M were used to evaluate the detection. In the evaluation stage, some of the Brontok and the normal exes were scanned against both B and M to determine each status.

The process of evaluation for every file is based on a simple rule indicated as follows:

```
String[] ngramX;  //ngrams of a given file to be evaluated
Foreach (String x in NgramX)
{
//compare against B and M for similarity statistical evaluation
int_counter_benign_exe =MatchAgainstB()
int_counter_malign_exe = MatchAgainstM()
}
//comparing the similarity statistic against a threshold
if (int_counter_benign_exe > counter_malicious_exe)
filestatus = true;
else
filestatus = false;
if(filestatus)
//displaying reputation ratings similarity against B and M
output(filename + “,”+ int_counter_benign_exe + "," + int_counter_malign_exe + "," + ((int_counter_benign_exe / totalmatch) * 100) +"% The file is good ");
else
output(filename + “,”+ int_counter_benign_exe + "," + int_counter_malign_exe + " ," + ((int_counter_malign_exe / totalmatch) * 100) +"% The file is bad ");
```

It begins with gathering the n-gram of the file to be evaluated. A list of this n-gram is then compared one by one against n-grams collections in B and M. The outcome upon completion of the test will be a similarity statistical value. This value will determine whether the file falls under a benign or malign category.

Figure 12 depicts the outcome when we scan a malware and it is accurately classified as malware. Figure 13 depicts the outcome of an evaluation against a normal file where the test accurately classifies the file as a normal file. Table 3 and 4 shows the overall scanning results of benign and all of the Brontok malware. We achieve 100% detection rate and accuracy on all files listed in Table 1 and 2.

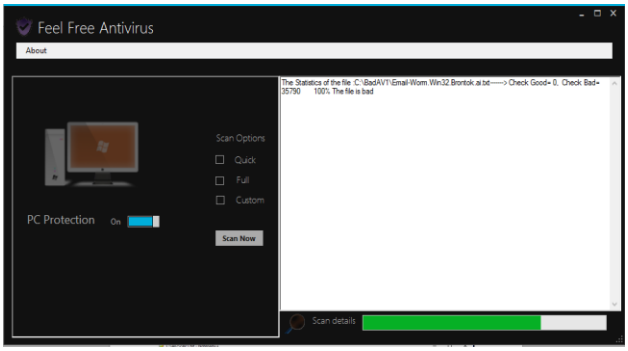


**Fig. 12:** Sample-scanning result for Brontok Email Worm.Win32.Brontok.ai.

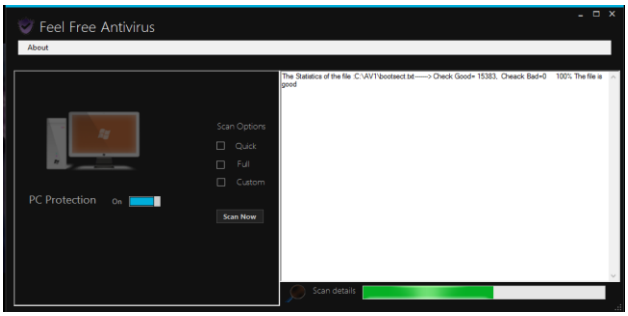


**Fig. 13:** Sample-scanning result for benign (normal bootsect.exe).

**Table 3:** Detection rate and accuracy of Brontok malware

| Detection Rate (%) TP/(TP+FN) | Accuracy (%) (TP+TN)/(TP+TN+FP+FN) |
|---|---|
| 100 | 100 |

**Table 4:** Detection rate and accuracy of benign programs

| Detection Rate (%) TN/(TN+FP) | Accuracy (%) (TN+TP)/(TN+TP+FP+FN) |
|---|---|
| 100 | 100 |

## 7. Conclusion

In general, this research was successfully completed, and all objectives were fully achieved. Lastly, the performance of the experiment was determined using the accuracy formula as shown in Table 3 and 4. From our experience, the most challenging part was the collection of the malicious sample (i.e. Raw Brontok family worm) and making sure that it did not infect the PC. Even if the PC was infected, we had to ensure that the infection was contained. Downloading the correct file is a huge challenge, as system crashes can occur when downloading an incorrect sample. There are many websites where one can download malwares, but few had the raw Brontok family worm with their original name. Tracking the right website for this malware family requires patience and time. Ensuring safety from accidental file execution or accidental corruption is also difficult, and this would end up to start from the beginning.

The limitation in this research was due to the extraordinary difficulty in getting the right source to download the malware sample. Furthermore, finding a source with sufficient Brontok family files to experiment with is another challenge. The effort involved in finding 24 samples caused an unfortunate limitation on the scope of the research.

Further research could be designed to evaluate a larger range of malware so that the results could be more representative. This may include not only executable but also script files, images, PDF, ransomware, etc. The reality is that a correctly executed test using the signature-based AV with n-gram technique could be extremely helpful to improve accurate detection. The implementation of some artificial intelligence techniques in the detection process is potential to provide interesting finding.

## Acknowledgement

This paper is supported by a research project funded under The Ministry of Higher Education, Malaysia (Grant No FRGS/1/2015/ICT01/USIM/02/1).

## References

[1] McDermid, J., 1989. ESEC'89: 2nd European Software Engineering Conference, University of Warwick, Coventry, UK, September 11-15, 1989. Proceeding (Vol. 387). Springer Science and Business Media.
[2] Emm, D., Garnaeva, M., Ivanov, A., Makrushin, D., & Unuchek, R. (2015). IT threat evolution in Q2 2015. Kaspersky Lab.
[3] Gianni. (2002), Brontok virus, http://no1tutorials.blogspot.my/2012/07/brontok-virus.html.
[4] Microsoft. Worm:Win32/Brontok.AR@mm. Windows defender security intelligence, https://www.microsoft.com/en-us/wdsi/threats/malware-encyclopedia-description?Name=Worm%3AWin32 %2FBrontok.AR%40mm
[5] F-Secure. (2010). Email-Worm: W32/Brontok.N, http://www.f-secure.com/v-descs/brontok_n.shtml.
[6] VX Heavens. Virus collection, http://vxer.org/.
[7] Georgia Tech Information Security Center. (2017). Open malware, http://openmalware.org/.
[8] Santos, I., Penya, Y. K., Devesa, J., & Bringas, P. G. (2009). N-grams-based file signatures for malware detection. ICEIS (2), 9, 317-320.
[9] Marhusin, M. F. (2012). Improving the effectiveness of behaviour-based malware detection. PhD thesis, University of New South Wales.
[10] VirusTotal, https://www.virustotal.com/#/home/upload.
[11] Zeltser, L. Malware sample sources for researchers, https://zeltser.com/malware-sample-sources/.