IJARSCT

ISSN (Online) 2581-9429

Impact Factor: 6.252

International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)

Volume 2, Issue 7, May 2022

Malware Detection Using Machine Learning Algorithms

Dr. Vajid Khan[1], Shriyansh Dubey[2], Sandeep Yadav[3], Vishal Gaikwad [4], Nishit Jambhulkar[5]

Professor, Computer Engineering, DPCOE, Pune, India[1]

Student, Computer Engineering, DPCOE, Pune, India[2,3,4,5]

Abstract: *In the present world, current antivirus software is only effective against known viruses if the malware contains new viruses, with signatures in place, it's hard to tell if it's malicious. Signature-based detection is less effective against zero-day attacks. Until the new hidden malware is detected it may spread in your computer system. This malware can exploit your system. According to research, malware has been found in the last 10 years it grew exponentially and caused significant economic losses to various organizations. Various antivirus companies are proposing solutions to protect against this malware attack. With the increasing speed, quantity, and complexity of viruses, malware poses new challenges to the antivirus community. The current state of research shows that researchers and antivirus organizations have recently begun to apply machine learning and deep learning techniques to analyse and detect various malwares. You can use machine learning techniques to create more effective antivirus software that can detect previously unknown and known malware, zero-day attacks, and more. In our project, we have proposed an approach that uses various machine learning methods and algorithms such as Vector Machine (SVM), Random Forest, and XGBoost.*

Keywords: Malware detection, virus, data mining, Information gain, random forest, machine learning, classification, enterprise, network, security.

I. Introduction

Malware, short for malicious software system, consists of programming (code, scripts, active content, and different software) designed to disrupt or deny operation, gather data that results in loss of privacy or exploitation, gain unauthorized access to system resources, and different abusive behaviour [1]. In a general term it is an outline to spread all sorts of hostile, intrusive, or annoying software system or program code. Software system is taken into account to be malware supported for perceived intent of the creator instead of any specific options.

Malware includes pc viruses, worms, Trojan, spyware, illegal adware, most rootkits, and other malicious and unwanted software system or program. In 2008, Symantec printed a report that "the unleash rate of malicious code and different unwanted programs is also surpassing that of legitimate software system applications." consistent with F-Secure, "As a lot of malwares made in twenty07 as within the previous 20 years altogether". While these could mean nothing to the typical home user, these statistics square measure atrocious keeping in mind the monetary implications of such threats imply to the enterprises just in case such threats penetrate and compromise the big volumes of information hold on and transacted upon. Since the increase of widespread net access, malicious software system has also increased and used for a profit, for examples forced advertising. Since 2003, the bulk of widespread viruses and worms are designed to require management of users' computers for contraband exploitation. Another class of malware, spyware, - programs designed to watch users' web browsing and steal non-public data.

Spyware programs don't unfold like viruses, instead square measure put in by exploiting security holes or square measure packaged with user-installed software system, such as peer-to-peer applications. Clearly, there's a really imperative ought to realize, not simply an appropriate methodology to find infected files, but also to build a sensible engine which will find new viruses by finding out the structure of system calls created by malware.

II. Current Antivirus Systems

An Antivirus are software which are used to prevent, detect, and remove any malware, including but not limited to computer viruses. A variety of strategies are typically employed by the anti-virus engines. Signature-based detection involves searching for known signatures of malware within executable code. However, computer systems may get affected by such viruses of which signatures doesn't exists. To counter such "zero-day" threats, heuristics can be used, that identify new viruses or variants of existing viruses by looking for known malicious code. Often, antivirus software can impair a computer's performance. Any decision taken hastily or incorrectly may lead to a security breach. If the antivirus software is of the type which employs heuristic detection than the success depends on achieving the right balance between false positives and negatives.

Today, malware may no longer be executable files. Traditionally, antivirus software has been heavily relied upon signatures of malwares to identify it but because of this of the newer kinds of malware, signature-based approaches are no longer effective. Although antivirus up to a certain extent can effectively contain virus outbreaks, for large enterprises, any breach could be potentially fatal. Virus makes are employing "oligomorphic", "polymorphic" and, "metamorphic" viruses, which encrypt parts of themselves and modify themselves as a disguise to not match virus signatures in the dictionary. Studies made in 2007 showed that the antivirus effectiveness of software had decreased drastically, particularly against unknown or zero-day attacks. Detection rates have dropped from 40-50% in 2006 to 20-30% in 2007. Independent research and studies on all the major virus scanners consistently show that no one provide complete 100% virus detection. The best ones provided are as high as 99.6% detection, while the lowest provided only 81.8% in tests conducted in Feb. 2010 [25]. Virus scanners can also produce positive results which may be false positive results as well.

III. Our Approach

As we've seen, current antivirus engine techniques aren't optimum in sleuthing viruses in real time they will be helpful in dominant viruses once they have infected systems, that is once more fearful for enterprises. This analysis is so aimed toward a central answer that works at the firewall level of the enterprise network. The entire system diagram is shown in and our method diagram is shown.

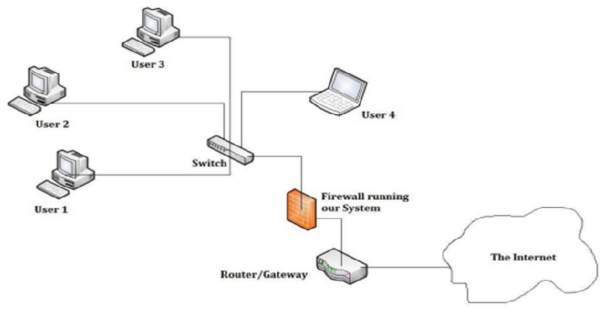


Figure 1: Network Diagram of the entire system

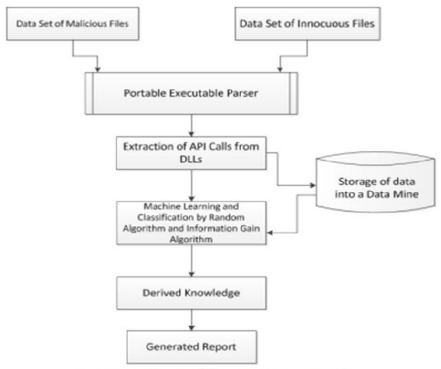

Data Set of Malicious Files
Data Set of Innocuous Files
Portable Executable Parser
Extraction of API Calls from DLLs
Storage of data into a Data Mine
Machine Learning and Classification by Random Algorithm and Information Gain Algorithm
Derived Knowledge
Generated Report

Figure 2: Process Diagram of our System

IV. Algorithm

The entropy of a variable X is defined as:

$$H(X) = -\sum_i P(X_i) \log_2(PX_i)$$

Where in H(P), the P(X) is as follows:

$$P(X_i) = \frac{Number \cdot of \cdot PE \cdot with \cdot x_i \cdot as \cdot certain \cdot API}{Total \cdot number \cdot of \cdot PE}$$

And the entropy of X after observing values of another variable Y is defined as:

$$H(X|Y) = -\sum_i P(Y_i) \sum P(X_i | Y_i) \log_2(P(X_i | Y_i))$$

The amount by which the entropy of X decreases reflects additional information about X provided by Y is called information gain, given by:

$$IG(X \mid Y) = H(X) - H(X \mid Y)$$

So, Machine learning is a branch of AI and a subject field involved with the design and development of algorithms that permit computers to evolve behaviours supported empirical knowledge, like from device knowledge or databases. A learner will cash in on data to capture characteristics of interest of their unknown underlying likelihood distribution.
Data is seen as examples that illustrate relations between ascertained variables. A major focus of machine learning analysis is to mechanically learn to acknowledge advanced patterns and make intelligent choices supported knowledge.
Further, we have a tendency to apply the Random Forest algorithmic rule (RFA) this can be a machine learning classification algorithmic rule to construct the classifier to find malware. A Random Forest may be a classifier that's comprised of a set of call tree predictors every individual tree is trained on a partial, severally sampled, set of instances hand-picked from the entire training set the anticipated output category of a categorified instance is that the most frequent class output of the individual trees.

V. Obtained Results

To determine whether or not our methodology will offer self-made results, we have a tendency to extracted knowledge from over5000 executables. These are a mixture of traditional and infected files. The first step was to form a hash map of all the executables and functions. After that the knowledge gain, algorithmic rule is employed to settle on solely the highest eightieth of the functions that are presumably to be present in harmful files. The knowledge gain is further corrected by victimization of this formula:

$$IG(X) = IG(X) \pm \left[\frac{\sum_{i=0}^{n} IG(Xi)}{n}\right]$$

This formula helps in correcting the error by adding or subtracting the typical data from the information gain data calculated. This is often almost like the error correction employing a commonplace deviation.
The results are shown below in figure 1.
After running Information Gain Algorithm, we get following functions which are shown in figure 2.
Then using this data, we run Random Forest Algorithm yielding following the results shown in figure 3.

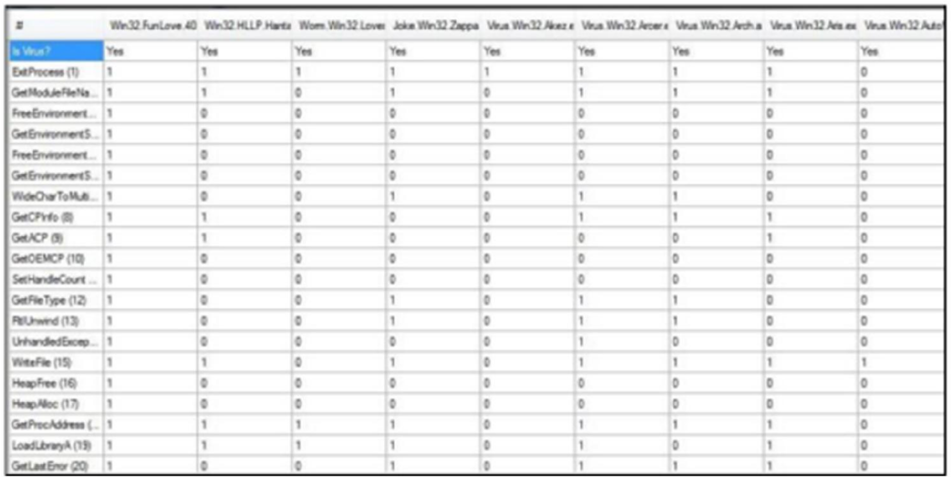
IJARSCT

ISSN (Online) 2581-9429

International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)

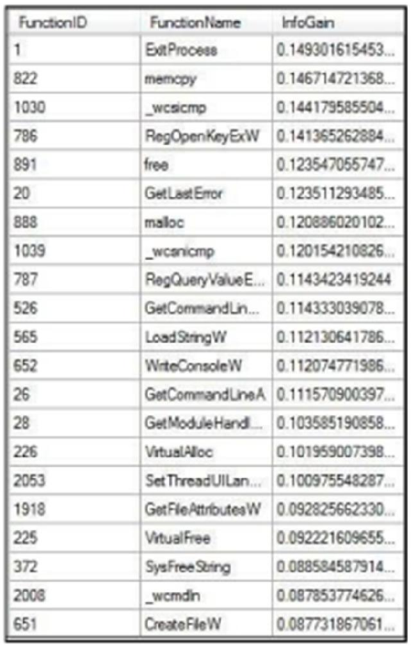
Impact Factor: 6.252

Volume 2, Issue 7, May 2022



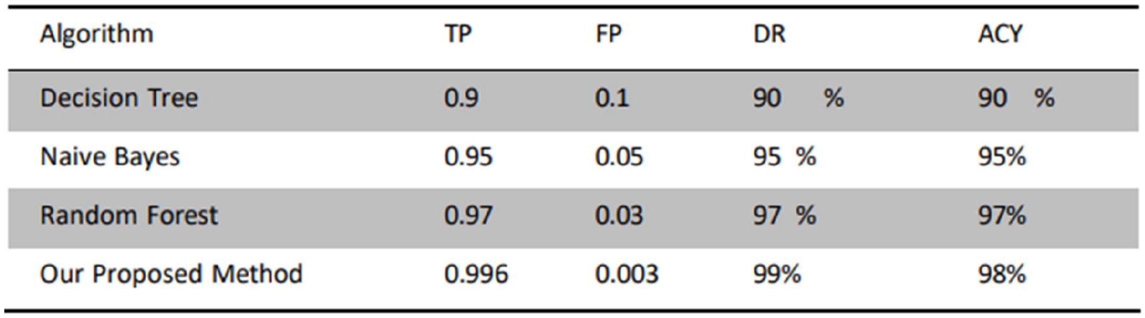
#	Win32.FunLove.40	Win32.HLLP.Hantz	Worm.Win32.Lover	Joke.Win32.Zappa	Virus.Win32.Akez.e	Virus.Win32.Arcer.x	Virus.Win32.Arch.a	Virus.Win32.Ata.ex	Virus.Win32.Auto
Is Virus?	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
ExitProcess (1)	1	1	1	1	1	1	1	1	0
GetModuleFileNa...	1	1	0	1	0	1	1	1	0
FreeEnvironment...	1	0	0	0	0	0	0	0	0
GetEnvironmentS...	1	0	0	0	0	0	0	0	0
FreeEnvironment...	1	0	0	0	0	0	0	0	0
GetEnvironmentS...	1	0	0	0	0	0	0	0	0
WideCharToMult...	1	0	0	1	0	1	1	0	0
GetCPInfo (8)	1	1	0	0	0	1	1	1	0
GetACP (9)	1	1	0	0	0	0	0	1	0
GetOEMCP (10)	1	0	0	0	0	0	0	0	0
SetHandleCount ...	1	0	0	0	0	0	0	0	0
GetFileType (12)	1	0	0	1	0	1	1	0	0
RtlUnwind (13)	1	0	0	1	0	1	1	0	0
UnhandledExcep...	1	0	0	0	0	1	0	0	0
WriteFile (15)	1	1	0	1	0	1	1	1	1
HeapFree (16)	1	0	0	0	0	0	0	0	0
HeapAlloc (17)	1	0	0	0	0	0	0	0	0
GetProcAddress (...	1	1	1	1	0	1	1	1	0
LoadLibraryA (19)	1	1	1	1	0	1	0	1	0
GetLastError (20)	1	0	0	1	0	1	1	1	0

Figure 1: Hash Map of Exe and API Functions



FunctionID	FunctionName	InfoGain
1	ExitProcess	0.149301615453...
822	memcpy	0.146714721368...
1030	_wcsicmp	0.144179585504...
786	RegOpenKeyExW	0.141365262884...
891	free	0.123547055747...
20	GetLastError	0.123511293485...
888	malloc	0.120886020102...
1039	_wcsnicmp	0.120154210826...
787	RegQueryValueE...	0.1143423419244
526	GetCommandLin...	0.114333039078...
565	LoadStringW	0.112130641786...
652	WriteConsoleW	0.112074771986...
26	GetCommandLineA	0.111570900397...
28	GetModuleHandl...	0.103585190858...
226	VirtualAlloc	0.101959007398...
2053	SetThreadUILan...	0.100975548287...
1918	GetFileAttributesW	0.092825662330...
225	VirtualFree	0.092221609655...
372	SysFreeString	0.088584587914...
2008	_wcmdln	0.087853774626...
651	CreateFileW	0.087731867061...

Figure 2: Information Gain Values of API Functions



Algorithm	TP	FP	DR	ACY
Decision Tree	0.9	0.1	90 %	90 %
Naive Bayes	0.95	0.05	95 %	95%
Random Forest	0.97	0.03	97 %	97%
Our Proposed Method	0.996	0.003	99%	98%

Figure 3: Experiment Results

IJARSCT

ISSN (Online) 2581-9429

Impact Factor: 6.252

International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)

Volume 2, Issue 7, May 2022

VI. Conclusion

In this analysis, we've projected a malware detection module supported by advanced knowledge mining and machine learning. Whereas such a technique might not be appropriate for home users, being very processor significant, this will be enforced at enterprise entry level to act as a central antivirus engine to supplement antivirus gift on user computers. This may not solely easily find proverbial viruses, however act as a data that may find newer kinds of harmful files. Whereas an expensive model requiring expensive infrastructure, will facilitate in protective valuable enterprise knowledge from security threat, and stop any financial damage.

References

[1] http://www.us-cert.gov/control_systems/pdf/undirected_attack0905.pdf

[2] "Defining Malware: FAQ". http://technet.microsoft.com. Retrieved 2009-09-10.

[3] F-Secure Corporation (December 4, 2007). "F-Secure Reports Amount of Malware Grew by 100% during 2007". Press release. Retrieved 2007-12-11.

[4] History of Viruses. http://csrc.nist.gov/publications/nistir/threats/subsubsection3_3_1_1.html

[5] Landesman, Mary (2009). "What is a Virus Signature?" Retrieved 2009-06-18.

[6] Christodorescu,M., Jha, S., 2003. Static analysis of executables to detect malicious patterns. In: Proceedings of the 12th USENIX Security Symposium. Washington.pp. 105-120.

[7] Filiol, E.,2005. Computer Viruses: from Theory to Applications. New York, Springer, ISBN 10: 2287-23939-1.

[8] Filiol, E., Jacob, G., Liard, M.L., 2007: Evaluation methodology and theoretical model for antiviral behavioral detection strategies. J. Comput. 3, pp 27–37.

[9] H. Witten and E. Frank. 2005. Data mining: Practical machine learning tools with Java implementations. Morgan Kaufmann, ISBN-10: 0120884070.

[10] J. Kolter and M. Maloof, 2004. Learning to detect malicious executables in the wild. In Proceedings of KDD'04, pp 470-478.

[11] J. Wang, P. Deng, Y. Fan, L. Jaw, and Y. Liu, 2003.Virus detection using data mining techniques. In Proceedings of IEEE International Conference on Data Mining.

[12] Kephart, J., Arnold, W., 1994. Automatic extraction of computer virus signatures. In: Proceedings of 4th Virus Bulletin International Conference, pp. 178–184.