Virus Detection And Prevention

Omkar Hatankar, Saee Petkar, Sandeep Kamble

Abstract— Our antivirus software basically works by scrutinizing program files against an extensive list of current signature/virus classified files; any program that matches a signature file is treated as suspicious and is either deleted or quarantined using Sandbox. An alert would appear to warn the user of the threat. For virus which are not known, it will adopt behavioral based detection to scrutinize suspicious program files. All executable files that enter the system go through this scan. Those that match the signatures are classified as viruses and are blacklisted. Benign files would directly run. Files which are not known will be kept separately in sandbox for analysis. User allowed files will be in the Whitelist.

Index Terms— Antivirus, Behavioral-based detection, Dynamic-based analysis, Sandbox, Signature-based detection, Static-based analysis, Virus.

———————————— ◆ ————————————

1 INTRODUCTION

A computer virus is a malware that, when executed, tries to infect other executable and alter their default behavior. A virus copies itself into an infected executable without permission or knowledge of a user. The first computer virus was a boot sector virus. Virus causes damage to system's file and operating system which comprises of system sectors, files, macros, companion files and source code [1]. Virus use internet as medium to spread throughout. The early detection of viruses is imperative to minimize the damages caused by them.

There are many antivirus defense mechanisms available today. These include signature-based and behavioral-based detection. The signature-based virus detection tools search all the files on a system for a signature. Code emulation creates a virtual machine and executes a virus on the virtual machine for detection. Once the virus is detected, it is no longer a threat. To bypass signature-based detection technique, virus writers have to create new viruses or change the existing viruses. Virus writers evade signature detection by generating metamorphic copies of a virus. Metamorphic viruses change their appearance while keeping the same functionality. Metamorphic viruses use different code obfuscation techniques to change the structure of the code. These techniques include code reordering through jumps, subroutine permutation, dead code insertion, equivalent instruction substitution, and rearrangement of instruction order (transposition). The statistical pattern analysis is the most successful technique to detect metamorphic viruses. In behavioral analysis, the behavioral characteristics of the executable is known as it is being observed in real-time, and inferences is made by an inductive decision algorithm on the threat level. All executables are treated as unknown, where it is up to the executable to prove it is acting in a safe, non-malicious manner. In doing so, the ability of detecting zero-day (unknown) attacks are substantially improved.

• *Omkar Hatankar is currently pursuing B.E. in computer engineering in Vidyalankar Institute Of Technology, India, E-mail: hatankaro@gmail.com*
• *Saee Petkar is currently pursuing B.E. in computer engineering in Vidyalankar Institute Of Technology, India, E-mail: saee.petkar@gmail.com*

2 PROBLEM STATEMENT

Virus is malicious bits of code that execute on the users' system causing leakage of information or damaging system. Viruses can also be used to create a network of botnets infected computers that can then be used as tools to host a Denial of Service attack. Currently, the antivirus software is equipped to recognize certain known viruses by checking the code of a file for these static signatures. The makers of these viruses often update virus code to prevent the detection of the virus through static signatures. Pre-defined static signatures in the antivirus program do not match to those of the virus. Thus, remaining undetected. Updates become necessary, but it is a slow process. Even then, some viruses are capable of modifying their code after each attack. Such metamorphic viruses cannot be detected by age-old antivirus software as signatures keep changing. Therefore behavioral-based virus detection technique is used.

3 PROPOSED SYTEM

Our antivirus system uses both the approaches- signature-based and behavioural-based virus detection.

1) Antivirus software searches for pre-ordered sequence of bytes within a file to identify the particular known virus.
2) The user may add a new signature if user finds it.
3) The behavior of the system is observed if virus detection is unsuccessful.
4) After positive virus detection, database of virus signatures is updated.
5) Harmful files are removed.

The antivirus has a predefined database of known signatures and hence while scanning, it creates the appropriate signature for each file (using MD5 or other hashes) and compares them with the predefined list. If they match, the file is treated as a 'threat' [2]. Database is updated when clicked on update button which provides you with a list of known signatures and adds it to the existing database thereby protecting you against latest threats. Our system has following process models:

3.1 Scanning Files

User will have options of scanning single file, folder or entire system.

3.2 Signature-based Virus Detection Technique

For every input file to system, we have used clam engine to extract signature of the file. This signature is then compared with stored signatures in a database.

3.3 Behavioral-based Virus Detection Technique

If extracted signature doesn't match with stored signatures, then behavior of system is scanned based on factors discussed later.

3.4 Update Database

After successful detection of virus with signature-based detection/ behavioral-based detection technique, database of stored signatures is updated with the addition of new detected virus signature.

3.5 Remove Process

The infected files are removed from the system.

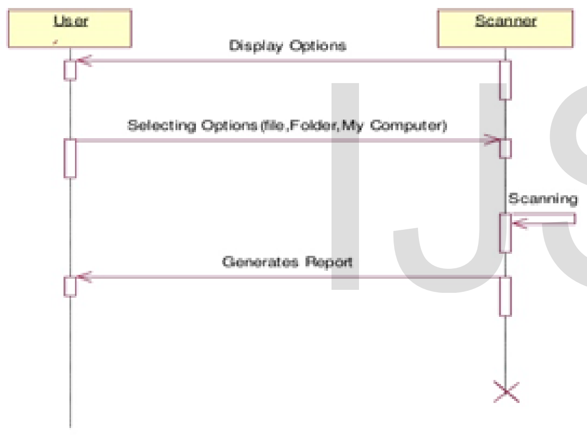

Fig. 1. Sequence Diagram: depicts the sequence of actions when our system is in use. Firstly, our system will display scanning options w.r.t. file, folder or user's entire system. Accordingly, user will select preferred option. System does the scanning and generates a report which shows whether user's system is safe or not.

4 System Overview

Our antivirus software will give user free choice of scanning system, folders and files. User has the privilege to update database of virus signatures manually. After behavioral detection, a report is generated of infected files. These files are either deleted or executed depending on their degree of infection to system.

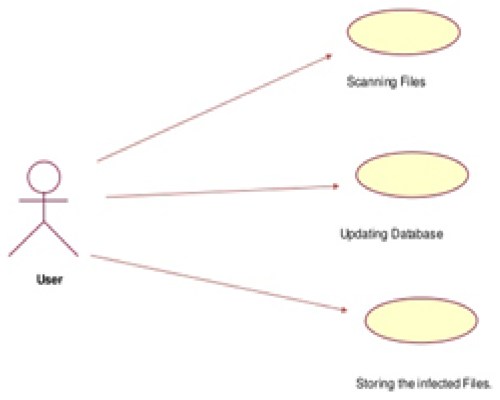

Fig. 2. Use Case Diagram: shows what all user can do. He can scan files for any malicious content, can update the signature database and can store the infected files in sandbox for isolation purpose.

5 Implementation Of Proposed System

Our system is client-server based architecture. Updating virus database is automatic. Static-based approach is initially used for detection of well-known viruses. If virus is not detected, then behavioral-based analysis is done using Support Vector Machine algorithm.

5.1 Static Based Analysis Detection

In signature analysis, code string patterns (signatures) are extracted from the target application's code and compared to a repository of known malicious code patterns. A virus signature is the fingerprint of a virus. It is a set of unique data, or bits of code, that allow it to be identified. It is referred to as a definition file or DAT file [3].

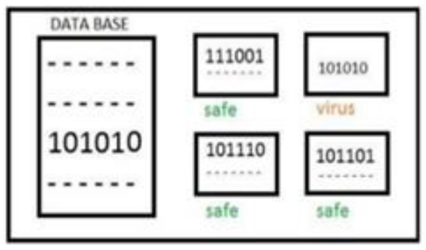

Fig. 3. Signature-Based Detection: shows that content of file is converted to binary and then compared with the database of stored signatures. Thus, the result shows whether the file is safe to use or not.

5.2 Dynamic Based Analysis Detection

Live monitoring of system is done for any malicious activities happening. Following are some common behavioral traits [4]:

1) *Payload persistence*

To ensure an attack is carried out to completion, it needs to persist across reboots and be able to resume upon starting. Common techniques used by ransomware includes placing a copy of its executable into the Windows startup directory, adding a registry run key entry or setting up a scheduled task, to name a few.

2) Anti-system restore

To ensure that any malicious actions cannot be undone, malware may try to disable system restore functionality. Ransomware for example, has been known to delete Windows shadow copies, which prevents encrypted data from being restored to an older unencrypted version.

3) Stealth techniques

Malware will try to execute in a stealthy manner to avoid being noticed by the user or detected by virus scanners. Common techniques include: injection into legitimate processes, executing from the %AppData% directory and using executables named the same as common Windows executables, to name a few.

4) Environment mapping

When malware is executed, it may map its system environment before initiating its setup procedure. This is typically done to determine if it's running on a real computer or on a sandbox environment that could be attempting to analyse it.

5) Network traffic

Ransomware that requires an internet connection, does so for two possible tasks: downloading of payload related files, and/or for the communication of the encryption key.

6) Privilege elevation

Executing malicious system-related activities may require access rights that are beyond those given to the victim's user account. For example, ransomware may want to overwrite the Master Boot Record, which can only be done as an Administrator. Simply asking for administrator access may work or other privilege escalation techniques may be used.

5.3 Support Vector Machine Algorithm in Supervised Learning

It is used for live monitoring of system. It gives optimized solution based on training. The untrained dataset has malicious and benign files, different behavioral characteristics observed in system and various classification schemes describing trained and predicted algorithms. In this algorithm, we plot each data item as a point in n-dimensional space (where n is number of features you have) with the value of each feature being the value of a coordinate. Then, we perform classification by finding the hyper-plane that differentiate the two classes very well.

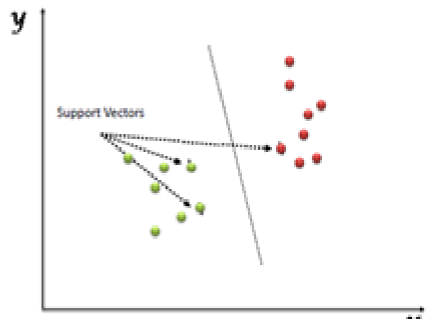

Fig. 4. Support Vector Machine (SVM): Displays hyper-plane separating harmful and harmless files. Red dots denote harmful files containing virus. Green dots denote harmless files.

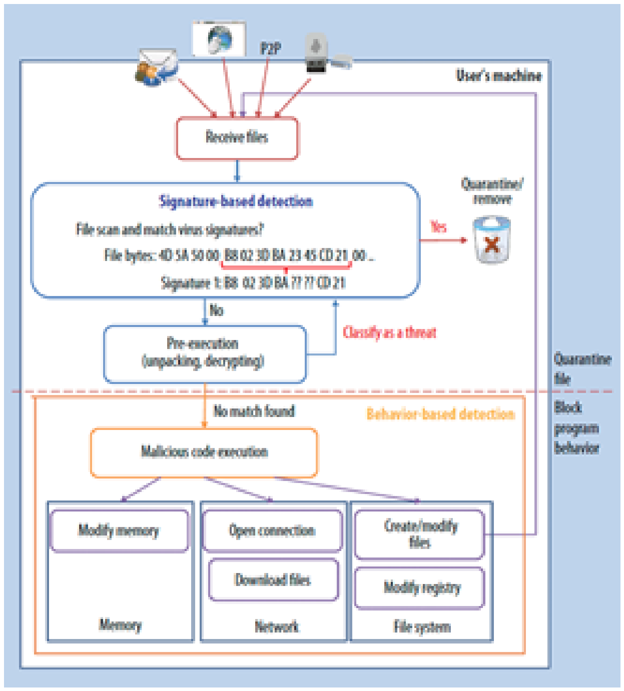

Fig. 5. Process Flow Diagram: describes the flow of our system. The received malicious files' content is compared with the database containing pre-defined signatures. If match found, then file is deleted else, it's observed under behavior-based virus detection; whether it modifies its memory size, or it performs any network based suspicious behavior or it modifies any file's content.

6 Conclusion

Our antivirus software can detect and remove known as well as unknown viruses. It takes less space and time to show the result. In future, we have decided to work more on behavioral traits. Quick detection of unknown viruses by supervised learning is to be implemented vigorously.

Acknowledgment

We are thankful to all those who helped us throughout the development of writing this paper. Their valuable and insightful inputs and constructive criticisms have been of utmost importance.

References

[1] Sudhakar Singh, P.K. Khare, J.M. Keller, P. Mor, M.K. Pathak, Sardar Patel College of Technology, Balaghat (M.P.), India, "Protection of Smartphone, Personal Computer and Other Similar Devices from Virus Infections", Vol. 2, Issue 10, October 2014.

[2] Neeraj Vats, Information Technology/ Cyber Security Professional, Techvella, "Ways to get rid of Malwares!!! How Does an AV Program saves systems from Viruses?", https://www.techwalla.com/articles/what-is-the-purpose-of-

antivirus-software, April 6, 2017.

[3] Computer Hope, “Virus Signature”, https://www.computerhope.com/jargon/v/virus-signature.htm, April 26, 2017.

[4] Daniel Nieuwenhuizen, “A behavioural-based approach to ransomware detection”, MWR Labs Whitepaper, labs.mwrinfosecurity.com, April 2017.