Network Based Anti-virus technology for Real-time scanning

Dr. Devara Vasumathi [1], Mr. T. Murali Krishna [2]

1. **Associate Professor, Department of Computer Science and Engineering, College of Engineering & Technology, Jawaharlal Nehru Technological University, Hyderabad, India**
2. **Lecturer, Department of Computer Science, College of Engineering & Technology, Jimma University, Jimma, Ethiopia**

Abstract

Negative aspects of networking lately became one of the most crucial problems in the information world. Protection against malware and malicious code must balance between being capable of fast detection of malware and being light on resources Since modern antivirus systems are not optimized in the terms of speed, this is one of the reasons why optimization towards performance and not only efficiency is very important. This article presents a method for real-time scanning with minimized load of the system and maximized precision to recognize the files needed to be scanned. Module, using rule-based method, for open source antivirus software was proposed and developed.

Keywords: *virus, malware, malicious code, real-time scanning*

1. Introduction

As more and more computers and other computing devices are interconnected through various networks, such as the Internet, computer security has become increasingly more important, particularly from invasions or attacks delivered over a network or over an information stream [13].

Malware has had a tremendous impact on computer systems and users. The rising number of computer security incidents since 1988 [4] suggests that malware is an epidemic [7]. Examples of this kind of code include computer viruses, worms, Trojan horses, and backdoors and others[3]. Malware code is handled by antivirus system, which techniques used for detecting malware can be categorized broadly into two categories: anomaly-based detection and signature-based detection. Figure 1 depicts the relationship between various types of malware detection techniques. Each of the detection techniques can employ one of three different approaches: static, dynamic, or hybrid (see Figure 1) [7].

Typically, a malware scanner spends the bulk of its time matching data streams against a large set of known signatures, using a pattern matching algorithm [15].

However, to perform any of these scan methods in real time scanning; detection system consumes a lot of CPU power, Which wouldn't be such a big problem with nowadays computers, but now, when a lot of PC operations goes to mobile world, mobile devices have severe resource constraints in terms of memory and power [16]. The same situation is in computers and notebook world –

Here are some new processors are introduced like Intel Atom, AMD Athlon Neo and Via Nano. These processors have less computing power, but also consume much less energy.

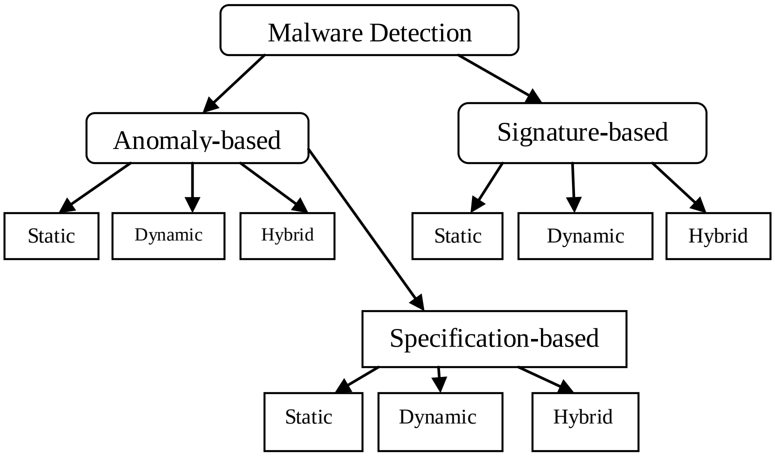


Figure 1 A classification of malware detection techniques [7]

It is important to mention, that to protect computer users, anti-virus engines have to use real-time protection methods. It means, that while file is not scanned and confirmed as clean, user cannot continue with his actions. Survey of M86 Security, which was presented in March 2010, demonstrates the importances of real-time scanning of the actual content users are accessing [1] on their mobile devices or desktop computers.

As everyone wants to be protected and secured from malware code (viruses, Trojan horses, worms etc.) and, at the same time, does not want to feel really having anti-virus software on his computer, in terms of resource consumption, this was thought of as a factor of utter importance while designing the rule-based methods of malware scanning. If unnecessary files that are sure not to be malware are skipped, the time and system resources are saved. But also it must be kept in mind that no infected files can be skipped as safe ones. The speed of operation is very important. Method, which determines, if the file should be scanned or not is suggested in this paper. In general it should answer the question “what to scan”. Rule-based detection of malware method was discussed in [12] and [18]. We have developed new rule-based method with different approach. In this paper we:

- Present a real time scanning engine combined with ClamAV library and signature data base.
- Present rule-based scan method which uses created real-time scanning engine.

The prototype was built for lab tests. Prototype engine with ClamAV [5] signature databases was launched on user computers.

2. Prior and related work

2.1 Types of malware and the importance of real-time protection

Malicious code could be described as "a generic term that encompasses viruses, trojans, spywares and other intrusive code." [17] As it is said in [7] - "The canonical examples of malware include viruses, worms, and Trojan horses." Why it is a real need of real-time scanning mechanisms when dealing with any of these kinds of malware will be explained in detail later in this article.

Virus is a computer code that replicates by inserting itself into other programs [7]. From definition of this malicious code type, it is clear that unless virus has access and ability to insert itself into program code, it is powerless. Thus the preferred method of protection against viruses is before it gains access to any components on the system. Real-time method offers such possibility.

As with worms, they replicate their code without inserting themselves into other programs, but via network connections, trying to infect not defended and open systems. Methods of rule-based real-time scanning make it possible to block the worm while it still has not infected possible host.

Trojan horses are performing unauthorized and possibly undesirable actions. Since they might affect the system so that their operation is almost invisible, it is important to stop their activity before their actions and not when consequences are already seen.

2.2 Real-time protection

There is no other good method to protect users from malware code without real-time protection engine. Real-time protection is also known as On-Access scan, Auto-Protect etc.

On-access scanning is an improvement over on-open, on-close, and on-exec scanning. An on-access scanner looks for viruses when an application reads or writes data, and can prevent a virus from ever being written to disk. Since scanning is performed only when data is read, as opposed to when the file is opened, users are not faced with unexpected delays [10].

A priori scanning for malicious code is the crucial factor when evaluating the performance of antivirus software, so different developers choose different methods for minimizing resource consumption. Some antivirus systems are set not to scan archive types or compressed files during real-time scanning thus reducing the load on the system, but this is not the best way out for it leaves unprotected (or partially protected) parts of the system. With growing number of objects in the system, number of potential threats is growing rectilinearly hence it becomes a difficult task for developers to design and develop antivirus that would manage to take number of actions needed to protect the system in real-time and be economical on resources. Having in mind the development and growing population of malicious code, this problem will not become less important with the growing capabilities of hardware, but will be one of the most important issues in mobile technologies, computers and equipment, prone to malicious activities.

Various benchmarks at [11] shows that some modern antivirus systems with real-time protection can prolong boot time up to 2 times, file compression, up to 1,5 times and extraction of archives up to 5(!) times. Such numbers talk for themselves and solid drop of performance on systems becomes a real problem that needs to be solved.

2.3 Rule based methods

It is known that experts can detect malware without antivirus, because they easily recognize suspicious behavior of software themselves, out of pure computer-based experience. Most of standard antivirus software use predefined malware signatures and usually consumes a lot of resources. Combining expert knowledge with signature based methods better performance can insure the same security level.

Real-time scanning for malicious code is important for it does not let malware to become a part of the system and hence protects it more than scanning that is being made post-factum. Real-time scanning can require vast amount of computer resources, so it must be optimized to be not only safe, but also light on system resources. For an effective defense, one needs virus-scanning performed at central network traffic ingress points, as well as at end-host computers. As such, anti-malware software applications scan traffic at e-mail gateways and corporate gateway proxies, and also on edge compute devices such as file servers, desktops and laptops. Unfortunately, the constant increase in link speeds, storage capacity, number of end-devices and the sheer number of malware, poses significant challenges to virus scanning applications, which end up requiring multi-gigabit scanning throughput. [15]

Quality of rule-based method decisions is a critical issue in majority of information-intensive businesses [2]. It concerns not only antimalware systems, but also monitoring systems [14] or automatic recognition of languages [8]. Having in mind the importance of antivirus software in such systems, quality of such decisions becomes even more important. In the context of antivirus and antimalware software, the quality of precise detection of malicious code is equal to evaluation of overall quality of service since the wrongful detection of malicious code might lead to undesirable consequences. E.g. if the rules are too strict, the chances of fake detection of malware are increased. The system must not be overloaded with yet another scanning mechanism that would increase consumption of resources, especially when talking about real-time scanning. Decisions of the method must be carefully

balanced. E.g. forward-chaining systems, though powerful if correctly designed and implemented, can become a devourer of resources if the problem is too large. The brute-force method of checking every rule
against every possible threat can start consuming resources in exponential expense.

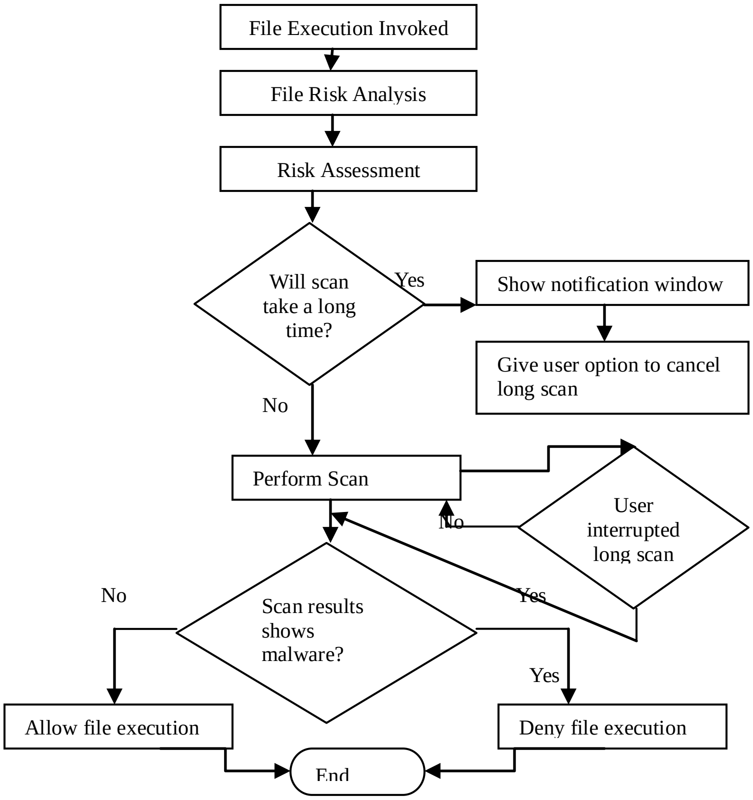


Figure 2. Scheme of patent [9]

Method and system for antimalware scanning are described in US patent, applied by Kaspersky lab [9]. In this invention, request for scanning a particular file has to pass a number of processes and only afterward the system is granted with access to it. It does not include scanning of large files – they are treated separately. The scheme of this patent is presented in Figure 2.

It is described what, when and how to scan for malware when virus signature database is renewed in US patent, applied by Microsoft [13]. Such aspects as decision whether new update is needed to be installed on the system or not, and a selective method of scanning files for malware are presented in this invention. The detailed scheme of the patent is presented in Figure 3.

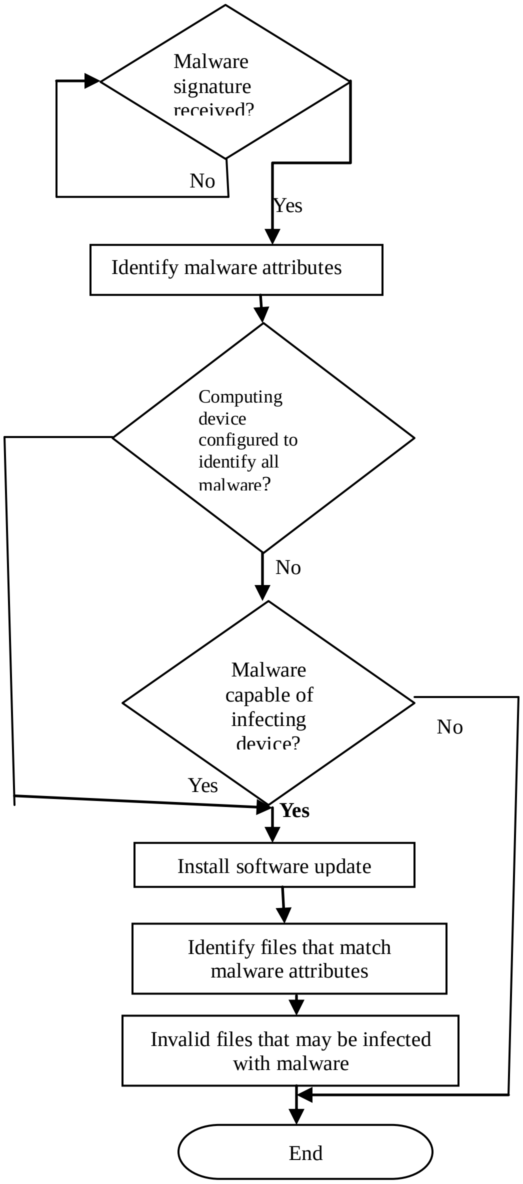

Malware signature received?
No
Yes
Identify malware attributes
Computing device configured to identify all malware?
No
Malware capable of infecting device?
No
Yes
Yes
Install software update
Identify files that match malware attributes
Invalid files that may be infected with malware
End

Figure 3. Scheme of patent [13]

Rule-based method for minimizing the worm damage in enterprise networks was presented in [15]. Proposed approach includes analyzing the parameters influencing worm infection, predicting the number of infected nodes by fuzzy decisions and optimizing the trust parameter in order to minimize the damage, made by fuzzy control.

In [18] authors present the malware detection algorithm that uses purely syntactic approach towards possible threats, ignoring the semantics of instructions. It handles a limited set of transformations of malware code used by hackers.

Above mentioned methods are not fully suitable for the real-time protection in the terms of system load management and precision of malicious code detection.

2.4 Clam AV based real-time scan engine

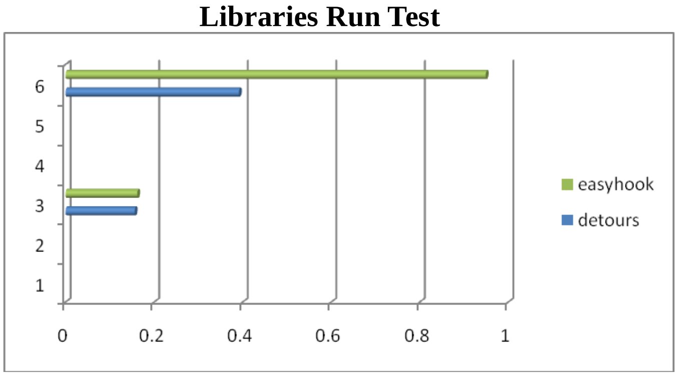
Two parts of real-time scanning engine are most important while defining the engine time performance: process hooking time and file scan time. This section will focus on hooking library time analyses and file scanning time.

Interception of Windows API functions requires deep understanding of the operating system since this process is associated with modification of memory processes of the system. The choice of used libraries is dependent on the requirements. The method for real time scanning (on-access scan) is chosen. Method should be able to take over functions that operating system uses for work with files ie. *Kernel32.dll* functions *CreateFile*, *OpenFile*, *ReadFile* etc. Also the speed of functioning system must be minimized and libraries must be written under GPL. In this test Detours and EasyHook libraries are chosen and used.

Detours library intercepts functions overwriting calling tables of processes. For every function, library rewrites two functions. Detours library can take over any function of any library.

Easyhook library supports interception of unmanaged source code while using managed code. It assures that no garbage of resources or memory are left in intercepted program. It is also able to use handlers to take over uncontrolled API functions.

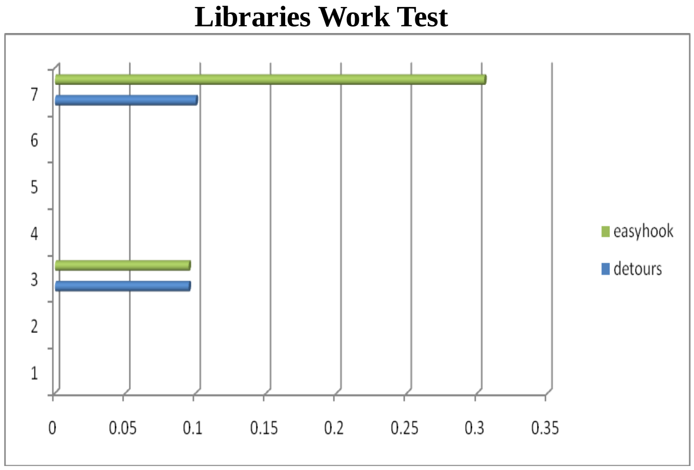
One of the prioritized qualities of our proposed method is speed thus these tests are very important when deciding which library to choose. The correct choice of library influences the overall speed of the system.



(X- axis: Time in ms)
figure 4. Time of library loading.

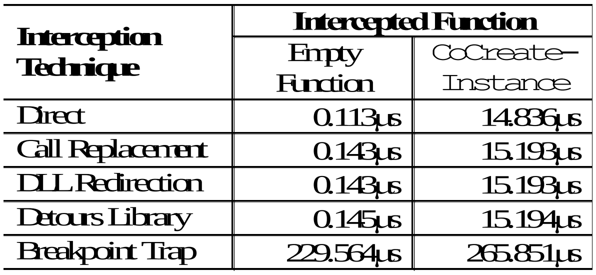
For each of the libraries, programs-servers were coded. They support command line format. Programs do not perform any tasks so that time, needed to complete tasks wouldn't influence the overall time. *Run* – application is loaded and waiting while new process will be finished. *Hook* – one of the libraries are inserted and takes over call of function *CreateFile*

From figure 4 it is clearly seen that when the program is started with EasyHook, it does it slower (142.27%). Next figure shows library work test. It can be seen that Detours library didn't have any effect on running time while EasyHook shows once again worse results. It was slower 305%



(X- axis: Time in ms)
Figure 5 Libraries work test

Next, the comparison of interception techniques and times from [6] is presented and compared with other API interception mechanisms.



Interception Technique	Intercepted Function	
	Empty Function	CoCreate-Instance
Direct	0.113µs	14.836µs
Call Replacement	0.143µs	15.193µs
DLL Redirection	0.143µs	15.193µs
Detours Library	0.145µs	15.194µs
Breakpoint Trap	229.564µs	265.851µs

Figure 6 Comparison of interception techniques

From these tests it is clearly seen how important it is to choose and design the access to resources carefully when using real-time scanning methods.

It was decided to use the freely available Clam AntiVirus (ClamAV) [5] scanner as the foundation for our real-time rule based anti-virus engine for desktop users. ClamAV consists of a core scanner library as well as various command line programs [10], but it does not have any real-time scanning method. So, we modified the ClamAV scanner library for use with our service and real-time protection.

ClamAV Virus Database as of February 2011 consisted of 879500 signatures. The ClamAV virus definition database contains two types of virus patterns: (1) basic patterns formed of simple sequence of characters that identify a virus, and (2) multi-part patterns that consist of more than one basic sub-pattern. To match a virus, all sub-patterns of a multi-part pattern must match in order. ClamAV virus patterns can also contain wildcard (*) characters. The combination of multi-part patterns and wildcard characters allows ClamAV to detect polymorphic viruses. Polymorphic viruses are more difficult to detect than non-polymorphic viruses, because each instance of a virus has a different footprint from other instances [10].

ClamAV uses a variation of the Aho-Corasick pattern-matching algorithm, which is well suited for applications that match a large number of patterns against input text.

The algorithm operates in two steps: (1) a pattern matching finite state machine is constructed, and (2) the text string is used as the input to the automaton [10].

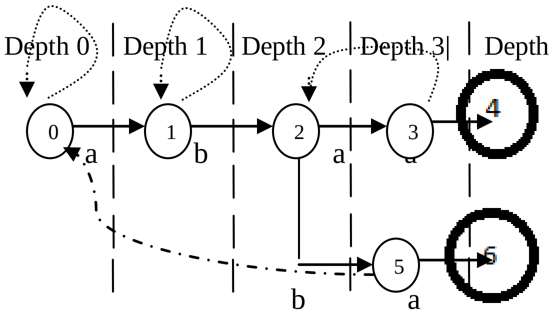


Figure 7. An automaton for keywords "abaa" and "abba" over the alphabet {a,b}. Success transitions are shown with solid lines. Final states are shown with bold circles. Failure transition are shown with dotted lines [10].

To construct a pattern matching automaton, the Aho-Corasick algorithm first builds a finite state machine for all of the patterns. Figure 7 shows the automaton for the keywords "abaa" and "abba" over the alphabet {a,b}. State 0 denotes the starting state of the automaton, and the final states are shown with bold circles.

First, the pattern "abaa" is added, creating states 0-4. Thereafter, the pattern "abba" is added, creating states 5-6. Only two additional states were required since both patterns share the same prefix "ab." Transitions over the characters of the patterns are called success transitions [10]

2.5 Rule-based method description

This method consists of list of rules which allows anti-virus system to avoid a lot of scan and signature matching work. Also some databases that are needed to store internal data as well as signatures used by ClamAV engine are used:

Main signature DB – a database provided by ClamAV anti-virus - the biggest malware database from ClamAV. It is being updated few times per year when all signatures from daily.cvd is moved to main.cvd database.

Daily signature DB – this ClamAV database contains newest malware signatures and is updated many times per day. Every few months all content of daily.cvd is moved to main.cvd.

File extension DB – it is the presented method's internal database which keeps extensions of files which should be scanned or not.

Standard file DB – database keeps standard operating system's files' checksums that are marked as trusted and should be never scanned.

Internal hash DB – for antivirus not to do the same job twice, the checksum of all scanned files is kept.

Most used files DB – as users' behavior is analyzed and most used files are stored, there must exist a database to store it.

2.6 Rules

Rule based method consist of a number of rules that allow us to get the needed result – decide what to scan. Every file has to go via all rules like shown in figure 8 in order to allow access to file or to check it with signature databases.

"File extension in extensions DB?" - this rule is one of the most important **-** once it decides the file should be scanned according to its extension. The rule is getting much more important these days because most of the threats come in standard executable files like ".exe" etc. The same tactics are described in US patent applied by Microsoft Corporation [13] in 2006. It describes how files should be picked up for scanning using only file extension database: "If a file is the type that may be infected by the malware, the file is scanned by antivirus software when a scanning event such as an I/O request occurs. Conversely if the file is not the type that may be infected by the malware and the file has not been previously identified as being infected, the file may be accessed without a scan being performed.".

"File size good for scan?" - rule is based on file size. Method decides to allow access to file according to files' size. System has adjustable file size, which can be changed to adopt to new malware threads. File size limitations comes from malware specifics – most of malware are not large files. As this rule has dangerous aspect – malware developers can make file which is big enough not to be scanned, file size should be updated with signature database updates.

"File located in standard file DB" - as all operating systems, as well as applications (like Microsoft Office) and services (e.g. database servers) has standard files that are compiled and linked by their producers, they can be marked as trusted and not scanned every time. This rule is very important, because most of the time users work with standard files. Some aspects of this rule are also described in US patent, applied by Kaspersky Lab in 2010 [9]: "As such when the software that is known to the anti-virus program (for example a previously installed copy of Microsoft Word) is launched, the anti-virus verification is relatively short for example limited to only virus signature checks of the dynamic linked libraries. On the other hand. When the software that is unknown to the anti-virus program is launched for the first time, more exhaustive antivirus checking can be performed".

"File located in internal DB" - rule checks if the file was scanned before and with which signature database version. In case the file is already scanned with newest database, it is not scanned again, and method allows access to the file. For file identification hash functions are used.

The method for antivirus checking proposed differentiates between the known executable files (i.e. files that the antivirus software has previously encountered in some sense on this machine) and unknown executable files[9].

"File scanned with newest Signature Dbs?" - as it is clear by now, that file has to be scanned, now it is very important to find out was it scanned with current version of virus signatures databases. In case it was – we can simple allow access to file without wasting expensive time to scan it again.

"File scanned with newest Main signature DB?" **-** If file was not scanned with newest virus signatures databases, it is even more important to determine if it was scanned with Main signature database. Such big importance comes from the fact, that Main data base has 846214 signatures (February 2011) [5], and Daily database has only 64700 signatures (February 2011) [5], which is about 13 times less, and means 13 times less CPU power. In case file was scanned with signatures from Main database it has to be scanned only with newest Daily database.

Principal scheme of the method is shown if figure 8. There are a couple of qualities that were most important while designing and creating the system. One of them was minimization of resource consumption. Another is correction of real-time scanning capabilities that would forbid malware to infect the computer.

System was tested in laboratory conditions. The system was installed on a desktop computer with Atom processor. Tests took one week and during that period of time, a constant smaller load on CPU was observed (10-15% less). During testing time, 15 newest different viruses were artificially presented to the system and all of these viruses were detected.

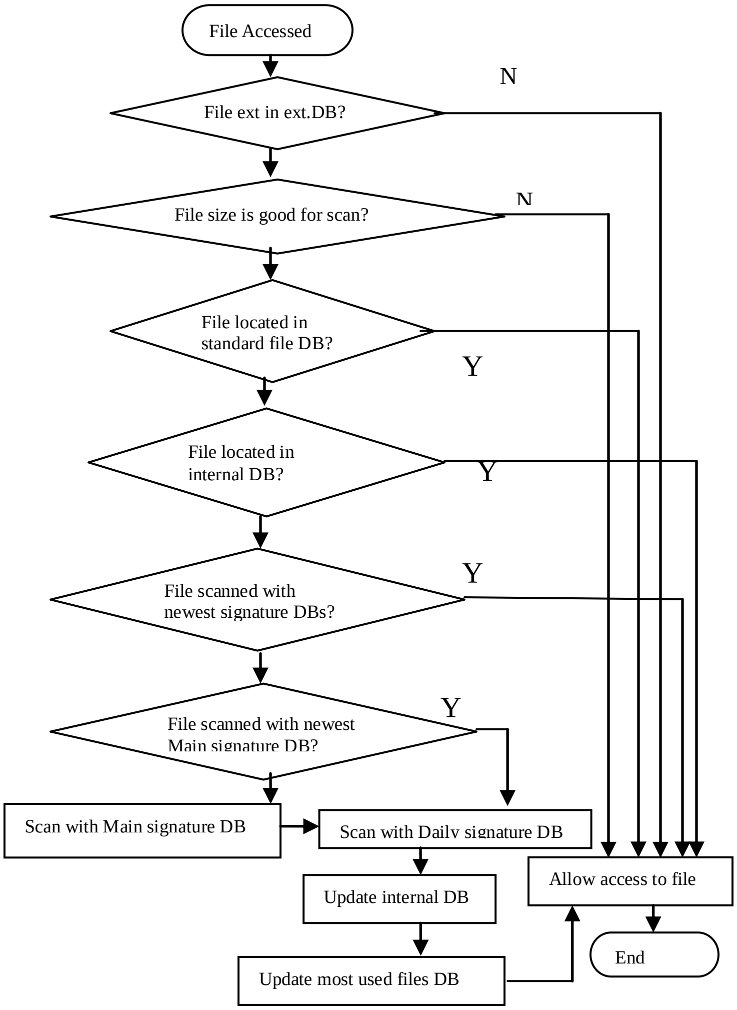

File Accessed
File ext in ext.DB?
N
File size is good for scan?
N
File located in standard file DB?
Y
File located in internal DB?
Y
File scanned with newest signature DBs?
Y
File scanned with newest Main signature DB?
Y
Scan with Main signature DB
Scan with Daily signature DB
Update internal DB
Update most used files DB
Allow access to file
End

Figure 8 Schema of the method

3. Future work

Method is still not completely adjusted and tested. To complete the work presented in this paper we need:

- To perform productivity tests and to compare the results with other rule-based methods and commercial anti-virus systems
- To test the effectiveness of the system using bigger collections of viruses.
- To apply the method for various states in different environments and during different time periods.

Conclusions

In this paper, we presented rule-based malware scanning method which can decrease consumption of computer resources without lowering security level.

Preliminary analysis shows that the system uses 10 to 15 percent less CPU resources. To achieve these results, analysis of real-time engine techniques was made.

It was shown that open-source ClamAV in combination with real-time scan engine and new rule-based method can perform all antivirus tasks on users computer.

References

[1] J. B. Anstis. Current Trends in Web Security Attacks and Best Practices to Stop Them, *virtual seminar*, 2010. http://www.m86security.com/webinars/04_10/Current_Trends_in_Web_Security_Attacks.pdf

[2] C. Cappiello, C. Francalanci, B. Pernici, A rule-based methodology to support information quality assessment and improvement, *Information Quality of' Studies in Communication Sciences'*, vol. 4, no. 2, December 2004, pp. 137-154.

[3] M. Christodorescu, S. Jha, J. Kinder, S. Katzenbeisser, H. Veith. Software Transformations to Improve Malware Detection *Journal in computer virology*, 2008, vol. 3, no. 4, 253-265

[4] http://www.cert.org/stats/

[5] Http://www.clamav.net

[6] G. Hunt, D. Brubacher, Detours: Binary Interception of Win32 Functions, *USENIX Windows NT Symposium, Seattle*, 1999.

[7] N. Idique, A.P.Marthur A Survey of Malware Detection Techniques. Tech. Rep. SERC-TR-286, Software Engineering Research Center (March 2007)

[8] J. Kapočiūtė, G. Raškinis Rule-Based annotation of Lithuanian text corpora, *Information Technology and Control*, 2005,Vol.34, No.3, 290-296

[9] Method and system for antimalware scanning with variable scan setting, United States Patent 7725941.

[10] Y. Miretskiy, A. Das, C. P. Wright, E. Zadok. Avfs: An On-Access Anti-Virus File System. *Proceedings of the 13th USENIX Security Symposium*, August 9-13, 2004, San Diego, CA, USA

[11] http://www.raymond.cc/

[12] S. Sanguanpong and U. Kanlayasiri, Worm damage minimization in enterprise networks, International Journal of Human Computer Studies 65(1) (2007), pp. 3–16

[13] System and method of caching decisions on when to scan for malware, United States Patent 7882561

[14] D. Strasunskas, S. L. Tomassen Scenario-driven information retrieval: supporting rule-based monitoring of subsea operations, 2007, Vol. 36, No. 1A, 87-92

[15] G.Vasiliadis, S. Ioannidis. GrAVity: A Massively Parallel Antivirus Engine. *Proceedings of the 13th International Symposium On Recent Advances In Intrusion Detection (RAID)*, September 2010, Ottawa, Canada.

[16] D. Venugopal, G. Hu. Efficient signature based malware detection on mobile devices, *Mobile Information Systems*, 2008, vol. 4, no.1, 33-49

[17] A. Vasudevan, R. Yerraballi. Spike: Engineering malware analysis tools using unobtrusive binary-instrumentation. In Proceedings of the 29th Australasian Computer Science Conference, pages 311–320, 2006.

[18] M. Christodorescu, S. Jha,S. A. Seshia et al.Semantics-Aware Malware Detection, Security and Privacy, 2005 IEEE Symposium, 32-46, 20

Authors Bibliography

1. Dr. DEVARA VASUMATHI received her M.Tech degree in Computer Science and Engineering from Jawaharlal Nehru Technological University, Hyderabad (JNTUH) in 2005. She received PhD in Computer Science and Engineering from JNTUH in 2011. Currently working as Associate Professor in CSE and also as Additional Controller of Exams, JNTU, Hyderabad, India. Her main research interests are Data Mining and Data Warehousing, Design Patterns, Web Technologies, and Operating Systems. She has got 11 years of teaching experience. She has published 11 research papers in various international journals. She is a life member in professional society of Indian Society for Technical Education (ISTE)

2. Mr. TELKAPALLI MURALI KRISHNA is pursuing PhD (CS) in Rayalaseema University, Kurnool, India. He received his M.Tech in CSE from Acharya Nagarjuna University, Guntur, AP, India in 2010. He received his M.Phil (CS) from Madurai Kamraj University, Madurai, TN, India in 2005. He received his MCA from Osmania University, Hyderabad, India in 1998. Currently working as a Lecturer in Computer Science, Jimma University, Jimma, Ethiopia. His main research interests are Data Mining, Software Engineering, Network Security, and Artificial Intelligence. He has 14 years of teaching experience. He has published 2 papers in international journals and presented 4 papers in national conferences. He is life member in CSI, IACSIT, IAENG, CSTA, and SDIWC. He is a member in reviewer board of IJCAT journal.