Using Case-Based Reasoning (CBR) for detecting computer virus

Abdellatif Berkat[1]
[1] Telecommunication Laboratory, Faculty of Technology, Abou-Bekr Belkaïd University
Tlemcen, 13000, Algeria

Abstract

The typical antivirus approach consists of waiting for a number of computers to be infected, detecting the virus, designing a solution, delivering and deploying a solution. In such a situation, it is very difficult to prevent every machine from being compromised by viruses. In this paper, we propose a new method, for detecting computer viruses, that is based on the technique of Case-Based Reasoning (CBR). In this method, (a) even new viruses that do not exist in the database can be detected (b) the updating of the virus database is done automatically without connecting to the Internet. Whenever a new virus is detected, it will be automatically added to the database used by our application. This presents a major advantage.

Keywords: *virus, antivirus, Intelligent systems, Case-Based Reasoning (CBR), detection of viruses.*

1. Introduction

Computer viruses are an omnipresent issue of information technology. A lot of books discuss their practical issues [5] or [21]. But, as far as we know, there are only a few theoretical studies on this topic. This situation is amazing because the term "computer virus" comes from the seminal theoretical works in the mid-1980's . We do think that theoretical point of view on computer viruses may bring some new insights to the area, as it is also advocated for example by *Eric Filiol* [20], an expert on computer viruses and cryptology. Indeed, a deep comprehension of viral mechanisms is from our point of view a promising way to suggest new directions on virus detection and defense [7].
Studies and researches show that a computer connected to the Internet may experience an attack every 39 seconds . New vulnerabilities in the system are discovered every few days. These vulnerabilities are fixed by the software vendors who provide patches and updates for the system. However, during this process the vulnerabilities are exploited by hackers, where malicious programs are installed on user machines to steal secret data for financial gains.
Unfortunately, our current ability to defend against new viruses is extremely poor and the basic approach of detection, characterization, and containment has not changed significantly over the last five years. The complexity of modern malware is making this problem more difficult. For example, *Agobot*, has been observed to have more than 580 variants since its initial release. Modern *Agobot* variants have the ability to perform denial of service attacks, steal bank passwords and account details, propagate over the network using a diverse set of remote exploits, use polymorphism to evade detection and disassembly [2].

Computer virus programmers use many techniques to avoid detection, such as space filling, compressing and encryption. On the another hand, the antivirus software tries to detect the virus by using various static and dynamic methods . However, all the existing methods are not sufficient to detect new viruses if the antivirus software is out-of-date. To develop new reliable antivirus software some new techniques must be exploited. Antivirus Smart are introduced at the end of this paper[2].

2. Antivirus techniques

2.1 Static Detection Methods

Static antivirus techniques attempt virus detection without actually running any code. This paragraph examines three static techniques: scanners, heuristics, and integrity checkers [5].

-Scanning method: Searches for sequence of bytes (strings) that are typical of a specific virus but not likely to be found in other programs.
-Heuristics Analysis: Heuristic analysis is an expert based analysis that determines the susceptibility of a system towards particular threat/risk using various decision rules or weighing methods. Multi-Criteria analysis (MCA) is one of the means of weighing.
-Integrity checkers: The integrity checker initially computes and stores a checksum for each file in the system it is watching. Later, a file checksum is recomputed and compared against to the original, stored checksum. If the checksums are different, then a change to the file occurs [5].

2.2 Dynamic Detection Methods

Dynamic antivirus techniques decide whether or not a code is infected by running it and observing its behavior [5]. The program monitors know the methods of virus activity including attempts to infect and evade detection. This may also include attempts to write to boot sectors, modify interrupt vectors, write to system files, etc... **.** For example, most virus activity eventually needs to call some system functionality, like I/O operations - only these actions have to be considered. No matter how obfuscated the I/O calls are statically, the calls will appear clearly when the code runs. Software monitors work best when the normal usage characteristics of the system are vastly different from the activity profile of an infected system. A virus might exhibit a dynamic signature like :
• Opening an executable, with both read and write permission.
• Reading the portion of the file header containing the executable's start address.
• Writing the same portion of the file header.
• Seeking to the end of the file.
• Appending to the file.

3. Case-Based-Reasoning (CBR)

Case-Based-Reasoning (CBR) has enjoyed tremendous success as a technique for solving problems related to knowledge reuse. Many examples can be found in the CBR literature . One of the key factors in ensuring this success is CBR's ability to allow users to easily define their experiences incrementally and to utilize their defined case knowledge when a relatively small core of cases is available in a case base [11].

The CBR process can be represented by a schematic cycle, as shown in Fig1. *Aamodt* and *Plaza* [1994] have described CBR typically as a cyclical process .

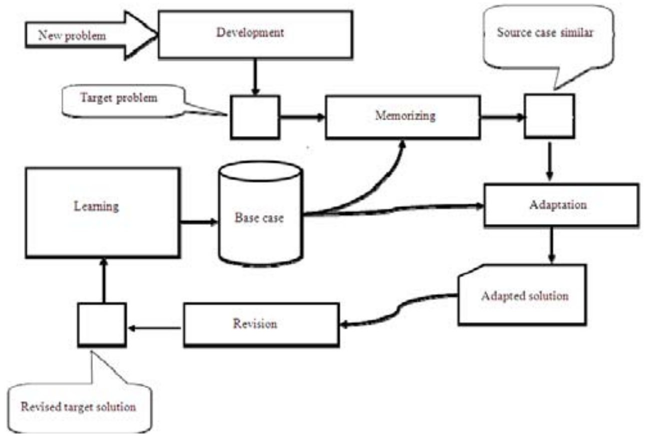


Fig1: CBR cycle

3.1 Development (Elaboration)
During the elaboration step, a request for experience reusing is triggered [14] .

3.2 Memorizing
In this step, the system calculates the similarities between the new case and stored cases through combining both component specific knowledge and general domain knowledge, and the components whose similarities surpass a threshold are returned [13].

3.3 Adaptation
Case adaptation is the process where a retrieved solution can be transformed into an appropriate one for the current problem . Several authors avoid this phase and prefer to develop the part of retrieval by considering that the case abundance will compensate adaptation task .However other authors consider adaptation as a crucial part of CBR systems because it confers to them their quality of problem solvers. Moreover, our goal is to propose a tool to the preliminary design stage , where the number of past experiences is limited and adaptation is therefore decisive [18].

3.4 Revision
When a case solution generated by the reuse phase is not correct, an opportunity for learning from failure arises. This phase is called case revision and consists of two tasks:
-Evaluate the case solution generated by reuse. If successful, learn from the success.
- Otherwise repair the case solution using domain-specific knowledge [1].

3.5 Learning
A very important feature of Case Based Reasoning is its coupling to learning. The driving force behind case based

methods has to a large extent come from the machine learning community, and Case-Based-Reasoning is also regarded as a subfield of machine learning [3]. Thus, the notion of Case-Based-Reasoning does not denote only a particular reasoning method, irrespective of how the cases are acquired but also a machine learning paradigm that enables sustained learning by updating the case base after a problem has been solved. Learning in CBR occurs as a natural byproduct of problem solving. When a problem is successfully solved, the experience is retained in order to solve similar problems in the future. When an attempt to solve a problem fails, the reason for the failure is identified and remembered in order to avoid the same mistake in the future.
Case-Based-Reasoning favours learning from experience, since it is usually easier to learn by retaining a concrete problem solving experience than to generalize from it. Still, effective learning in CBR requires a well worked out set of methods in order to extract relevant knowledge from the experience, integrate a case into an existing knowledge structure, and index the case for later matching with similar cases [1].

4. Using (CBR) for detection computer virus

From all the methods that we have seen previously already used by major corporations in the fight against viruses (scanning, heuristics, spectral, monitor behavior, etc..), we propose another method used in several areas of artificial intelligence called 'Case-Based-Reasoning'.

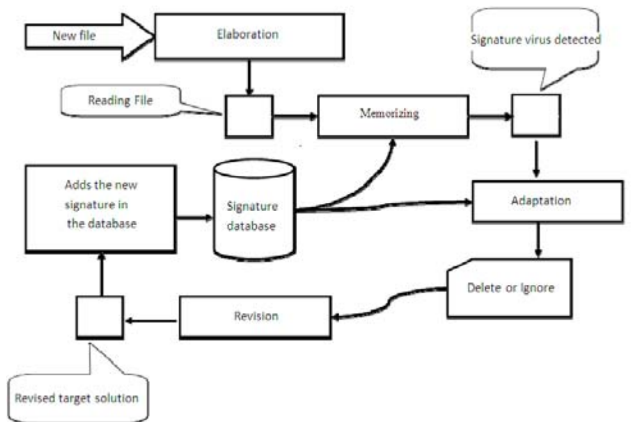


Fig2: Steps of our application

4.1 Development (Elaboration)
At this stage, our system (application) will load, in its main memory, a new file to analyze.

4.2 Memorizing
In our case, the memorization phase enables to test if any source problems (signatures of viruses already detected) appear in the file being analyzed. If a part or the whole signature (more than two characters of the four in the signature) are detected, the file will be considered as an infected file. For example:
Assume that the string '**vir1**' is a signature of a virus recorded in the database. If the analysis of this file finds the word '**Vir2** ', then this file is assumed to be infected.

4.3 Adaptation
Example:
We assume that a file has been infected by a virus having the signature 'vir1' and was removed by the system. If a new file is infected by another virus having the signature '**Vir2**' (similar to '**vir1**') then the proposed solution is: *Delete*. This is called derivational adaptation (deriving the target solution from the source).

4.4 Revision
During this phase, the user has the choice to accept or reject the solution proposed by the system.
Therefore if, for instance, the proposed solution is *Delete* , then after the revision phase the user can accept this solution and remove the file or can ignore this option.

4.5 Learning (Adds the new signature in the database)
Here, if the new virus signature is not in the system case base, it is added with its proposed solution.

5. Application Description
In this section, we present the different functions used in our application and an example of the implementation.

5.1 Programmed functions

5.1.1 Void Find_files ( )
This function is used to search the files '***. COM**' (i.e. MS-DOS files). It fills a table of strings (called StringGrid1) with the names and sizes (in bytes) of files found.

5.1.2 Void Read_File (String F_name)
This function opens the file **'f_name'** read-only, transforms it in hexadecimal, browses the base case (base of virus signatures) and calls for the function **'Find (String f_name, Sig String, String fl)** ' and then for the function **'UpDate_Bdd ()'**.

5.1.3 Void Find (String f_name, String Sig, String fl)
This function searches for the signature that is in the variable 'Sig' in the text **'fl'** of file **'f_name'**. If found (i.e. the file is infected), this function calls for the function **'Supp_fl'** to delete the file **'f_name'**. If it finds three of the

four characters of the same signature, it calls for the function '**Ajout_Sig**'.

5.1.4 void Ajout_Sig (String F_name, String Sig)

This function tests whether the new signature in the variable 'Sig' is in the case base. If yes, it does nothing. If not, it prompts the user to delete the file '**f_name**'. If he accepts, the function **'Ajout_Sig'** removes the file and adds the signature in a temporary vector **'New_Sig'** (vector of new signatures).

5.1.5 Void UpDate_Bdd ()

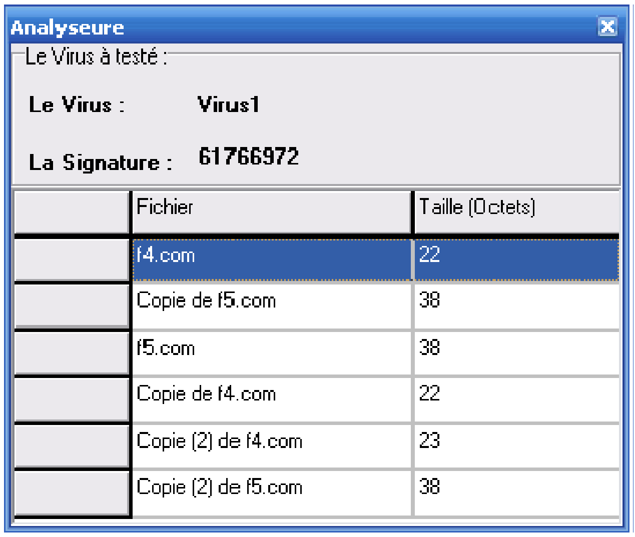
If there are new signatures in the vector **'New_Sig'**, the function **'Void UpDate_Bdd ()'** puts them at the bottom of the case base with the virus name '*Unknown* ' .

5.1.6 Void Supp_fl (String FileName)

This function deletes the file '**FileName**'.

5.2 Running the application

To program this application we need:
-The assembly programming language (which we used to create test viruses).
-The C + + programming language.

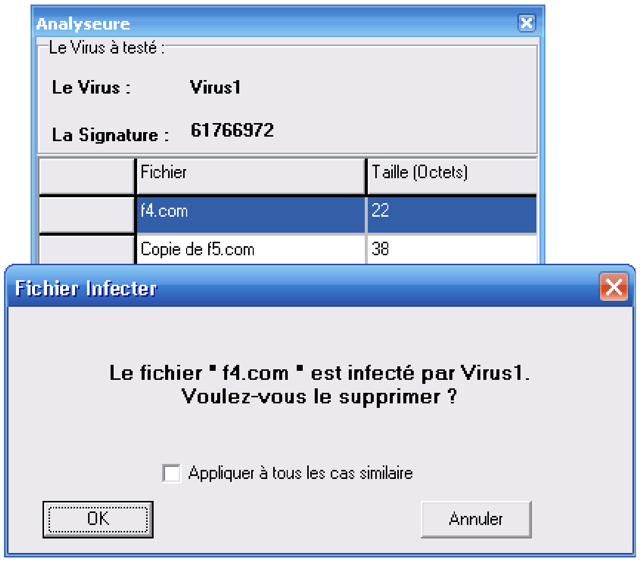


Analyseure
Le Virus à testé :
Le Virus : Virus1
La Signature : 61766972

	Fichier	Taille (Octets)
	f4.com	22
	Copie de f5.com	38
	f5.com	38
	Copie de f4.com	22
	Copie (2) de f4.com	23
	Copie (2) de f5.com	38

Fig3: Running the application

The messages and application windows are programmed in the French language.
The analysis is automatically done when running the application. which is programmed solely to analyze the '***. COM**' executable files.
The main window shows the name and the signature of the detected virus with infected files and the size of each file.



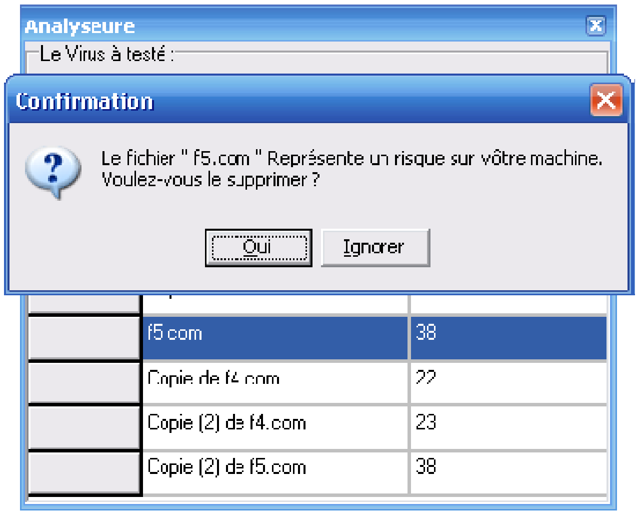
Analyseure
Le Virus à testé :
Le Virus : Virus1
La Signature : 61766972

	Fichier	Taille (Octets)
	f4.com	22
	Copie de f5.com	38

Fichier Infecter
Le fichier " f4.com " est infecté par Virus1.
Voulez-vous le supprimer ?
Appliquer à tous les cas similaire
OK Annuler

Fig4: detection of a virus existing in the database

When it detects a virus reported in the database, a message will be displayed in French, meaning **''filename' file is a risk' for your machine. Do you want to delete?'**. We can either *accept* or *cancel*. We can apply this decision to all similar cases by selecting '**Apply to all similar cases ' .**



Analyseure
Le Virus à testé :

Confirmation
Le fichier " f5.com " Représente un risque sur vôtre machine.
Voulez-vous le supprimer ?
Oui Ignorer

	f5.com	38
	Copie de f4.com	22
	Copie (2) de f4.com	23
	Copie (2) de f5.com	38

Fig5: detecting a new virus

When it detects a new virus the message **'the file 'filename ' is a risk for your machine. Do you want to delete?'** is displayed. Here, we have the choice to *accept* or *ignore* .

6. Conclusions

File analysis has the same principle of the scanning method. Some of the advantages of this method are the detection of new viruses, learning how to place new virus signatures in the database that can be automatically updated.
This study, which is based on artificial intelligence, has never been carried out by any research teams working in the antivirus field. It enabled us to get closer to a worldwide domain of research that affects all our activity sectors today.

References

[1] A. Aamodt, E. Plaza (1994); Case-Based Reasoning: ‘’Foundational Issues, Methodological Variations, and System Approaches’’. AI Communications. IOS Press, Vol. 7: 1, pp. 39-59.
[2] Essam Al Daoud, Iqbal H. Jebril and Belal Zaqaibeh,’’ Computer Virus Strategies and Detection Methods’’, Int. J. Open Problems Compt. Math., Vol. 1, No. 2, September 2008.
[3] Jacob Ziv,Fellow ,IEEE ,and Abraham Lempel,member IEEE,’’A Universal Algorithm for Sequential Data Compression’’,IEEE Transactions on information Theory, VOL. IT-23, NO. 3, MAY 1977 .
[4] Hazem Y. Abdelazim(Assistant Professor, IT Department, College of Computer Science), Khaled Wahba(Assistant Professor, Systems and Biomedical Engineering Department Faculty of Engineering),’’ System Dynamic Model for Computer Virus Prevalance’’ , Cairo University .
[5] John Aycock(University of Calgary Canada) ,Book’’ Computer Viruses and Malware’’, Library of Congress Control Number: 2006925091 .
[6] Kevin W. Hamlen, Vishwath Mohan, Mohammad M. Masud,Latifur Khan, Bhavani Thuraisingham(Computer Science Department, University of Texas at Dallas, 800 W. Campbell Rd., Richardson, Texas 75080, USA) ‘’Exploiting an Antivirus Interface’’, Preprint submitted to Elsevier ,April 21, 2009 .
[7] G. Bonfante, M. Kaczmarek, and J.-Y. Marion(Loria, Calligramme project, B.P. 239, 54506 Vandoeuvre-l`es-Nancy C´edex, France, and ´Ecole Nationale Sup´erieure des Mines de Nancy, INPL, France.),’’ Abstract Detection of Computer Viruses’’, inria-00115368, version 1 - 21 Nov 2006 .
[8] John Aycock, Rennie deGraaf, and Michael Jacobson, Jr.(Department of Computer Science University of Calgary Calgary, Alberta, Canada)’’ Anti-Disassembly using Cryptographic Hash Functions’’,TR 2005-793-24.
[9] Danilo Bruschi, Lorenzo Martignoni, Mattia Monga,(Dipartimento di Informatica e Comunicazione Universit`a degli Studi di Milano),‘’Using Code Normalization for Fighting Self-Mutating Malware’’, Comelico 39/41, 20135 Milano – Italy .
[10] Madihah Mohd Saudi(National ICT Security & Emergency Response Centre(NISER) Malaysia) Shaharudin Ismail(Islamic University College of Malaysia (KUIM) Malaysia),, ‘’An Efficient Control of Virus Propagation’’ .
[11] Christina Carrick and Qiang Yang(Simon Fraser University ,Burnaby, BC, Canada, V5A 1S6) , Irene Abi-Zeid and Luc Lamontagne(Defense Research Establishment Valcartier Decision Support Technology 2459, boul. Pie XI, nord Val Belair, Quebec, Canada, G3J 1X5),’’ Activating CBR Systems through Autonomous’’, Springer-Verlag Berlin Heidelberg 1999 .
[12] Magda liliana RUIZ ORDONEZ(Girona university),‘’ Multivariate statistical process control and case-based reasoning for situation assessment of sequencing barch reactors’’,ISBN:978-84-691-6833-2,Diposit legal:GI-1299-2008 .
[13] Mingyang Gu, Agnar Aamodt and Xin Tong, ‘’Component retrieval Using Conversational Case-Based Reasoning’’, Department of Computer and Information Science, Norwegian University of Science and Technology, Sem Sælands vei 7-9, N-7491, Trondheim, Norway +47 7359 7410 .
[14] Amélie Cordier, Bruno Mascret, Alain Mille,’’ Extending Case-Based Reasoning with Traces’’,Université Lyon 1, LIRIS, UMR5205, F-69622, France, March 30, 2009 .
[15] Juan Corchado, Brian Lees, Colin Fyfe, Nigel Rees and Jim Aiken ,’’Neuro-Adaptation Method for a Case-Based Reasoning System’’, Computing and Information Systems, Vol 5, No. 1, p.15-20 .
[16] Francisco J (Dpto. Informática y Automática – Universidad de Salamanca). García, Juan M(Dpto. Informática y Automática – Universidad de Salamanca), Corchado and Miguel A. Laguna(Dpto. Informática – Universidad de Valladolid),’’CBR Applied to Development with Reuse Based on Mecanos’’ .
[17] Peter J.Funk(DepartmentofComputer Engineering),‘’Reuse, Adaptation and Validation of System Development Processes’’, Mälardalen University.
[18] Eduardo Roldan,Stéphane Negny,Jean Marc Le Lann ,Guillermo Cortes‘’Constraint Satisfaction problem for Case-Based Reasoning Adaptation :Application in Process Design’’,20th European Smposium on Computer Aided Process Engineering _ESCAPE20,2010 Elsever B.V.
[19] Jean-Marie Borello, Éric Filiol, Ludovic Mé,’’ From the design of a generic metamorphic engine to a black-box classification of antivirus detection techniques’’, Springer-Verlag France 2009 .
[20] Eric Filiol ,Book’’ Computer viruses: from theory to applications’’, Springer-Verlag France 2005 .
[21] Mark A. Ludwig,Book:‘’The Little Black Book of Computer Viruses’’, American Eagle Publications, Inc,1996 .

M. Abdellatif BERKAT was born in Algeria in 1987. He obtained his Master’s Degree in Telecommunications, from Abou Bekr Belkaid University, Tlemcen, Algeria, in 2010. M. Abdellatif BERKAT is interested in the following topics: antenna design, algorithmic and programming theories, optimization algorithms, development of artificial intelligence methods. M. Abdellatif BERKAT is a doctorate student in the same university working on antenna design.