Metamorphic computer virus detection by Case-Based Reasoning (CBR) methods

Abdellatif Berkat[1]

[1] Telecommunication Laboratory, Faculty of Technology, Abou-Bekr Belkaïd University
Tlemcen, 13000, Algeria
Berk.abdellatif@gmail.com

Abstract

Metamorphic virus employs code obfuscation techniques to mutate itself. It absconds from signature-based detection system by modifying internal structure without compromising original functionality.

In this paper, we propose a new method, for detecting metamorphic computer viruses, that is based on the technique of Case-Based Reasoning (CBR). In this method:

-Can detect similar viruses with high probability.

- 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

Metamorphic computer virus , antivirus , intelligent systems , Case-Based Reasoning (CBR) , detection of viruses , detection technique .

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 defence [24].

Computer virus programmer uses many techniques to transform their virus to avoid detection, such as, polymorphic and metamorphic are specifically designed to bypass detection tools.

In this paper, we are interested by a Metamorphic Virus , such as , Metamorphic Virus can reprogram itself. it use code obfuscation techniques to challenge deeper static analysis and can also beat dynamic analyzers by altering its behaviour, it does this by translating its own code into a temporary representation, edit the temporary representation of itself, and then write itself back to normal code again. This procedure is done with the virus itself, and thus also the metamorphic engine itself undergoes changes.

DOI : 10.5121/ijsea.2011.2401

Metamorphic viruses use several metamorphic transformations, including Instruction reordering, data reordering, in lining and outlining, register renaming, code permutation, code expansion, code shrinking, Subroutine interleaving, and garbage code insertion. The altered code is then recompiled to create a virus executable that looks fundamentally different from the original. For example, The source code of the metamorphic virus *Win32/Simile* is approximately 14,000 lines of assembly code. The metaphoric engine itself takes up approximately 90% of the virus code, which is extremely powerful. *W32/Ghost* contains many procedures and generates huge number of metamorphic viruses; it can generate at least 3,628,800 variations [8].

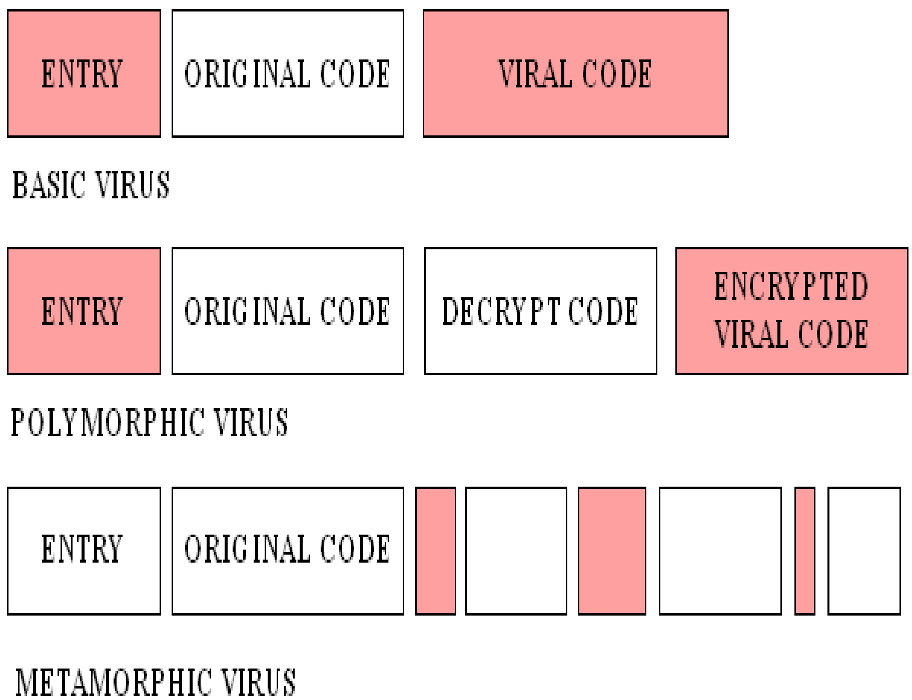

Figure1: Three kinds of viruses [4]

2. Related work

As metamorphic viruses employ complicated techniques, many different methods have been developed to detect metamorphic viruses. Each detection method has its own pros and cons. Some of the detection techniques are highlighted below [7] .

-Geometric Detection technique relies on “shape heuristic”; this allows to find whether a file is infected, or not, by learning the file structure of the virus and looking for learnt structures in the infected files. Often, this technique is prone to false positives as it simply learns the layout of the virus and does not learn about the virus at the instruction level.

-Code emulation is employed by creating a virtual machine which emulates the underlying hardware including processor, memory, and peripherals and runs an operating system. This technique detects viruses by running suspicious files on its guest virtual machine and looks for any malicious activities and patterns. The above technique has the ability to detect complicated viruses but it needs considerable system resources to create a virtual machine [7] .

As our research is focused on using Case Based Reasoning (CBR) for metamorphic virus detection, CBR will be discussed in the following section. And a Smart Antivirus are introduced at the end of this paper.

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 .

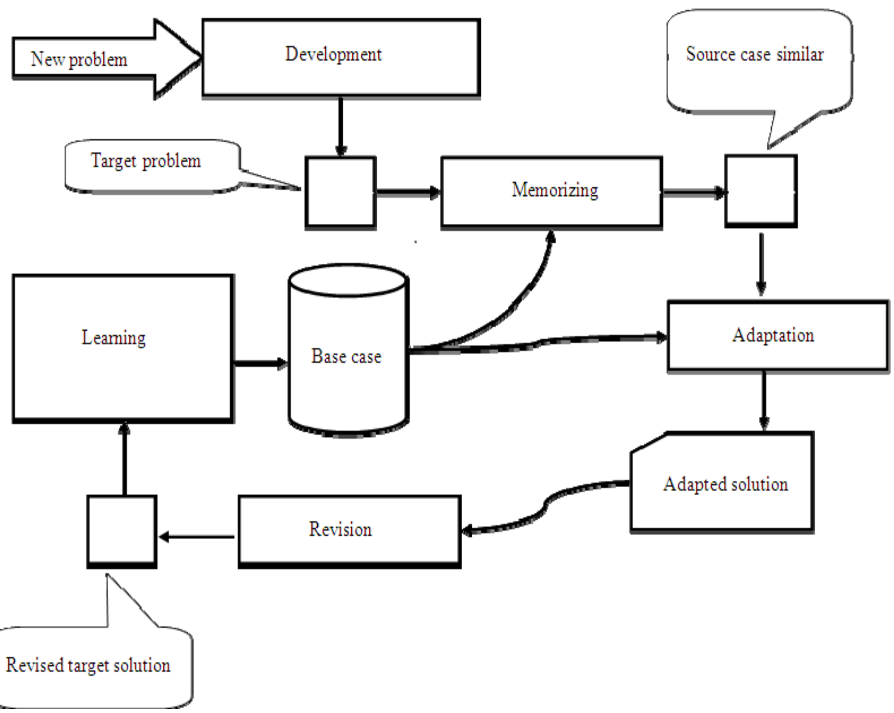

Figure2: CBR cycle

3.1 Development (Elaboration)

During the elaboration step, given a target problem, a request for experience reusing is triggered [14], retrieve from memory cases relevant to solving it. A case consists of a problem, its solution, and, typically, annotations about how the solution was derived.

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 by product 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 behaviour, etc..), we propose another method used in several areas of artificial intelligence called 'Case-Based-Reasoning'.

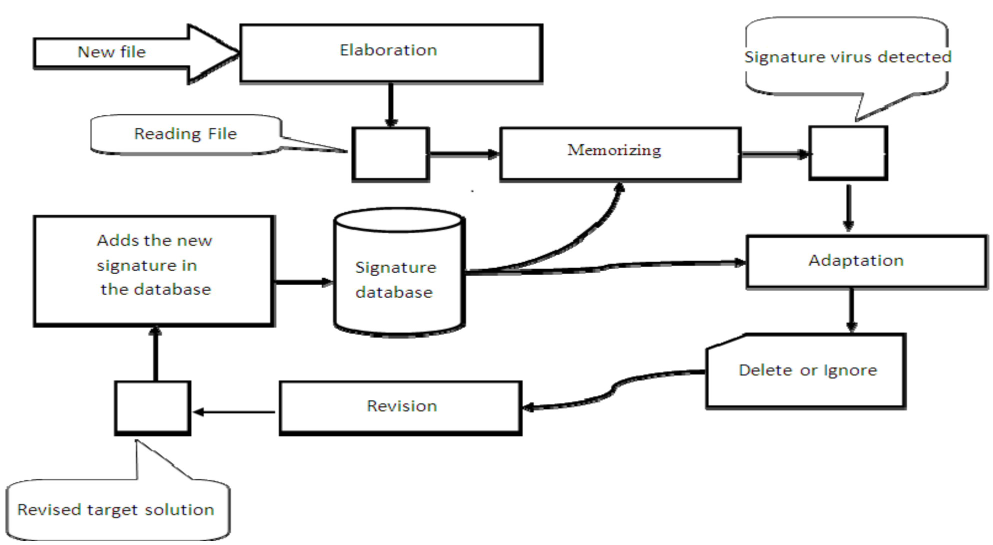

Figure3: Steps of our application

4.1 Development (Elaboration)

This stage, presenting the first process executed, such as, 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) is 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)

A new case is created with the problem situation and its solution, as proposed by the system. Future follow-up contacts will modify information in this case depending upon the reported results of the solution. Since this case is memorized, it will be possible for the system to reuse it in a subsequent problem solving episode [25].

Here and in our application, if the new virus signature is not in the system case base, it is added with its proposed solution in the database.

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 *'filename'* read-only, transforms it in hexadecimal, browses the base case (base of virus signatures) and calls for the function *'Find (String filename, Sig String, String fl)* ' and then for the function *'UpDate_Bdd ()'*.

5.1.3 Void Find (String filename, String Sig, String fl)

This function searches for the signature that is in the variable *'Sig'* in the text *'fl'* of file *'filename'*. If found (i.e. the file is infected), this function calls for the function *'Supp_fl'* to delete the file *'filename'*. 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 *'filename'*. 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)

If the function *'Void Find (String filename, String Sig, String fl)'* detects any signature that is in the variable *'Sig'* in the text *'f1'* of file *'FileName'* then 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.

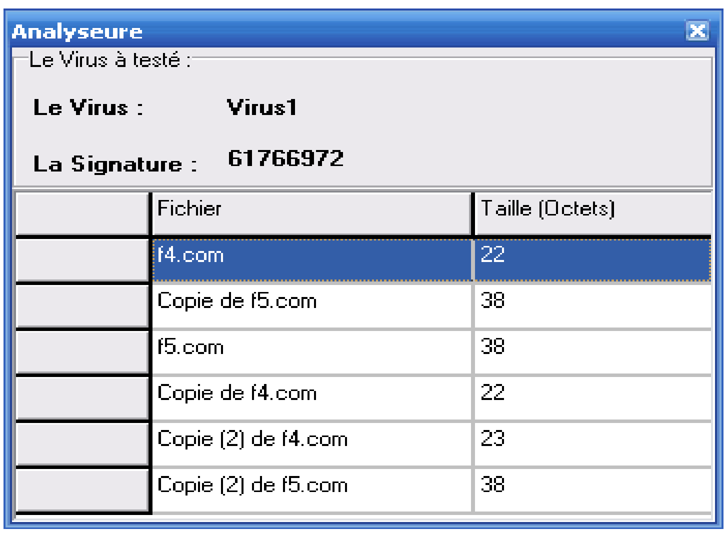

Figure4: Running the application

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.

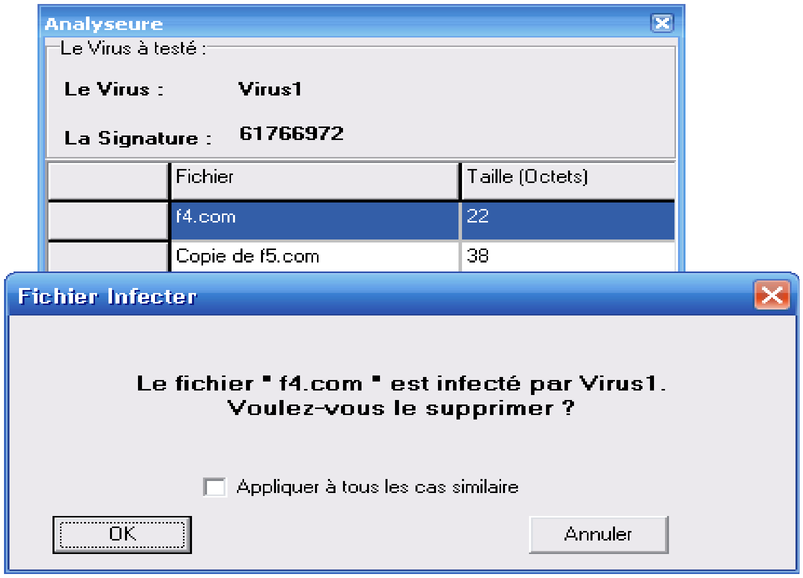

Figure5: detection of a virus existing in the database

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 '* **.**

Figure6: 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

Metamorphic viruses are in a sense advanced polymorphic viruses: on each replication, the code to be executed completely mutates, without altering its functionality. Thus, encryption is not anymore necessary and, when used, the decryption method as well as the decrypted code of the virus is different for each new generation.

In other , the antivirus software trying to detect the viruses by using variant static and dynamic methods . However; all the existing methods are not adequate. To develop new reliable antivirus software some problems must be fixed. This paper suggested new smart procedures to detect the metamorphic viruses by using case base reasoning (CBR) methods.

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.

7. Future Work

To more thoroughly evaluate the performance of the CBR approach, it would be useful to test on a larger set of virus variants and also test on different types of viruses. Ideally, we would like to find viruses that are similar to normal programs to a degree that the similarity index alone cannot distinguish the viruses from normal code. Only with such data can we evaluate the effectiveness of the CBR approach to detecting metamorphic viruses. However, it appears that no metamorphic kit available today is capable of producing such challenging viral code .

In other, we trained our models on disassembled virus executables. The disassembling process can take some time and the results depend on the quality of the disassemble . To speed up virus pre-processing and to eliminate the reliance on a particular disassemble, we could attempt to train the CBR directly on the binary code of the viruses. Other machine learning techniques, such as data mining might also work directly on the binaries.

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] RUO ANDO, NGUYEN ANH QUYNH,YOSHIYASU TAKEFUJI, ‘’ Resolution based metamorphic computer virus detection using redundancy control strategy’’, Graduate School of Media and Governance, Keio University, 5322 Endo Fujisawa, Kanagawa, 252 Japan.

[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] Sharmidha Govindaraj, ‘’ Practical Detection of Metamorphic Computer Viruses’’, The Faculty of the Department of Computer Science,San Jose State University, December 2008 .

[8] Srilatha Attaluri,''Detection Metamorphic Virus using profile HIDDEN MARKOV models'', The Faculty of the Department of Computer Science,San Jose State University ,December 2007 .

[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 .

[22] Hitesh Tahbildar(Department of Computer Engineering and Application, Assam Engineering Institute, Guwahati, Assam 781003, India), Bichitra Kalita(Department of Computer Application, Assam Engineering College, Guwahati, Assam 781003, India), ''HEURISTIC APPROACH OF AUTOMATED TEST DATA GENERATION FOR PROGRAMS HAVING ARRAY OF DIFFERENT DIMENSIONS AND LOOPS WITH VARIABLE NUMBER OF ITERATION'', International Journal of Software Engineering & Applications (IJSEA), Vol.1, No.4, October 2010 .

[23] Dr.R.SatyaPrasad(Dept.of Computer science &Engineering, Acharya Nagarjuna University, Nagarjuna Nagar-522524), O.NagaRaju(Dept.of Computer science &Engineering, Nagarjuna University, 522524), Prof.R.R.LKantam(Dept.of Statistics, Acharya Nagarjuna University,

Nagar-522524),’’ SRGM with Imperfect Debugging by Genetic Algorithms’’, International Journal of Software Engineering & Applications (IJSEA), Vol.1, No.2,April 2010 .

[24] 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 .

[25] Isabelle Bichindaritz, Emin Kansu, and Keith M. Sullivan ,’’Case-Based Reasoning in CARE-PARTNER: Gathering Evidence for Evidence-Based Medical Practice’’, Clinical Research Division ,Fred Hutchinson Cancer Research Center ,1100 Fairview Avenue N., D5-360 ,Seattle, Washington 98109-1024 .

Author

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. Abdellatif BERKAT is interested in the following topics: antenna design, algorithmic and programming theories, optimization algorithms, development of artificial intelligence methods. Abdellatif BERKAT is a doctorate student in the same university working on antenna design.