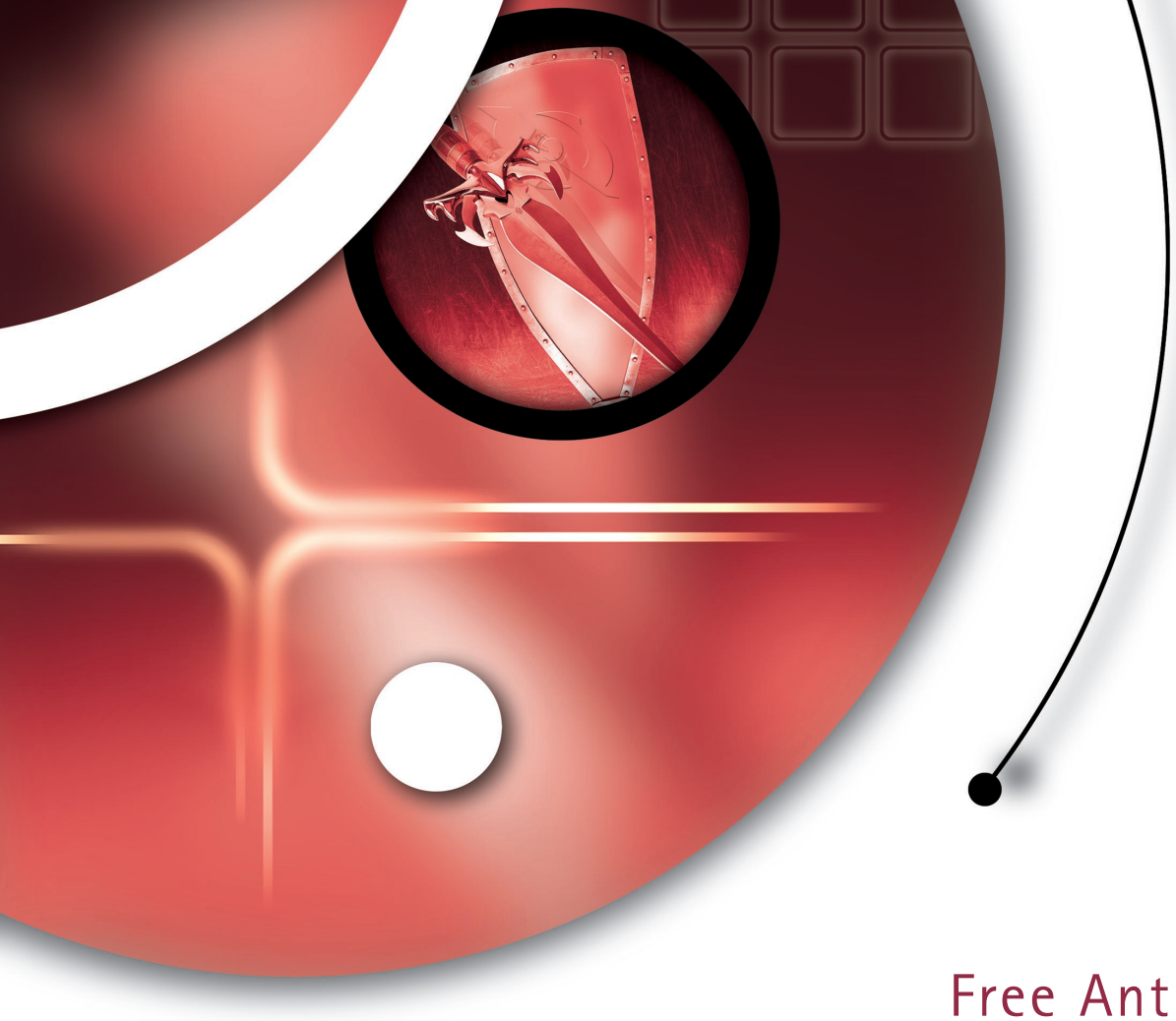

Free Anti-Virus Tips and Techniques

Common Sense Methods to Protect Yourself from Macro Viruses

By Chengi Jimmy Kuo
Senior Fellow, NAI Labs
Director Anti-Virus Research,
Anti-Virus Emergency Response Team (AVERT)

network ASSOCIATES
Who's watching your network

WHITE PAPER

Table of Contents

Introduction

In the summer of 1995, the Word Concept virus was unleashed upon an unsuspecting world, changing the scope of the virus problem forever. For the first time, viruses could reside in common word processing and spreadsheet documents. Since that time, Word and Excel macro viruses have become the most dominant virus threat to organizations and individuals alike, appearing at a rate of over 200 new viruses per month. "In 1999 alone, $7.6 billion in damage was done by viruses, many of them macro viruses affecting Word and Excel environments."

Network Associates offers a comprehensive solution for all existing and new macro viruses in its McAfee Total Virus Defense suite scanning engine. Part of any good virus security policy, however, involves things you can do at no cost to reduce your exposure to macro viruses.

The objective of this paper is to introduce a variety of free macro anti-virus techniques and discuss the pros and cons of each. This paper concludes with the author discussing several methods he personally uses to protect himself against macro viruses in his day-to-day work.

AUTHOR PROFILE

Chengi Jimmy Kuo is a well-known anti-virus researcher, specializing in how viruses affect the common user and how best to protect users' data. He is the author of numerous technical papers which have been independently translated into several languages, and he has appeared on TV and in newspapers worldwide thanks to his virus expertise. Kuo holds a Bachelor of Science degree in Engineering and Applied Sciences from the California Institute of Technology, and his previous areas of research include IBM PS/2 BIOS (his initials can be found in the some models of BIOS chips), AIX-UNIX development, and natural language processing. Kuo presently serves as Director of Anti-Virus Research for the Network Associates Anti-Virus Emergency Response Team (AVERT), and holds a Senior Fellow position at NAI Labs, a Network Associates advanced research facility.

Introduction to Macro Viruses

Macro viruses take advantage of "macro" utility tools built into programs such as Microsoft Word and Excel. By riding on data files exchanged through the application, macro viruses infect great numbers of documents. While macro viruses are application-specific, they are not operating system dependent, which means they can quickly travel in e-mail, downloads, floppies and groupware applications.

Macro viruses can exist in any number of products that give a user the ability to write macro scripts that can, in turn, write to the disk to propagate more macros. Because of its widespread usage, the product with the most macro viruses today is Microsoft Word. Viruses spread easiest in the MS Word environment because documents can contain both text and macros. By combining both text and macros, the user has much more power and usability features. The two go hand in hand. More power to the user. More potential for macro viruses.

Microsoft Excel is similarly afflicted. Excel macro viruses began appearing some time after the original Word macro virus, but are being discovered at an alarming rate today. The same dynamics that affect Word also affect Excel. Excel also uses OLE2 container files, combining macros and all of the cell functions and data in the same file.

When Microsoft Office 97 was released, all macro languages converged upon Visual Basic 5 (VB5), making cross-application viruses a theoretical possibility. This makes the possibility of macro viruses for other platforms only a matter of time, especially as more and more vendors independently support VB5.

Viruses are defined by their ability to spread. A Word macro virus spreads easiest when it intercepts the macro execution path by making use of one or more of Word's AUTO macros, or by using a menu replacement. Then it places itself into the global environment by, for instance, updating the normal.dot file. Most of the approaches below target that scenario.

Early Anti-Virus Suggestions from Microsoft

After the Concept. A virus first appeared in mid-1995, Microsoft responded promptly with several suggestions to help reduce the risk of infection. Although these early attempts proved ineffective once the scope of the macro virus problem became apparent, they do provide an interesting insight into Microsoft's initial attempts to solve the Concept. A problem.

Create a Payload Macro

One early suggestion supplied by Microsoft was to create a Payload macro to counteract the effects of Concept.A. While potentially effective against Concept.A, this method would have no effect on any of the thousands of other macro viruses in existence. With thousands of macro viruses in circulation today, Microsoft no longer recommends this approach.

PROS: Early solution that protected against Concept.A

CONS: No protection against other viruses

ScanProt

ScanProt is a macro package written by Microsoft. Its original intention was to protect against Concept.A and to provide a mechanism for users to be alerted if any incoming document contained macros. One unintended side effect of the original ScanProt workaround was the fact that various ScanProt macros were actually absorbed into spreading viruses. This caused a "mating" scenario between an existing virus and ScanProt, producing a new virus variant. The positive benefits of this approach, however, have since been incorporated directly into Word 7.0a and Word 97.

Another action undertaken by ScanProt is to rename macros associated with known viruses to alternative names. Although this approach makes some known viruses non-functional, it also makes them irremovable by some anti-virus products. With more than 200 new macro viruses appearing each month, it is also an ineffective method of maintaining ongoing virus security.

PROS: Early solution that protected against Concept.A
Alerts if any macros exist in document (now a feature of Word 7.0a and 97)

CONS: ScanProt macros can be absorbed and spread as part of new viruses

Prompt to Save Normal Template

This is an option available from within Word. Microsoft made this option the first macro virus prevention method when it suggested its use against Concept.A. To activate the

option, simply click on Tools and Options..., then choose the Save tab. From this menu, check the Prompt to Save Normal Template option.

As noted above, viruses spread by definition. This is most easily accomplished by getting into the global environment. The global environment is represented by the file normal.dot. If a virus attempts to alter normal.dot and this option is in use, Word will inform you that there is a request to change the Normal Template as you attempt to exit. At this time, you can respond that you do not wish to allow such a change. Presumably, the user would know of any intentional changes. An unexpected attempt to modify this file could, therefore, indicate a virus attack in progress.

Even with this option in use, however, it is possible to open an infected file and infect the environment. The warning does not occur until exit from Word. Thus any documents opened and saved after the initial infected document will also be infected. Furthermore, many of today's most prevalent macro viruses are aware of this attempt, and can easily deactivate this feature themselves, rendering it largely ineffective.

PROS: Easy to set
Commands required to set this feature can be automated
Reduced risk from early macro viruses like Concept.A, Wazzu.A, and NPad

CONS: Easy for a "smart" virus to deactivate
Does not inform until AFTER infection when user exits Word

Microsoft Word Macro Viruses

The Shift Key

Most macro viruses make use of Word's AutoOpen and AutoClose macros to operate. Disabling these built-in macros can be an effective way of stopping such macro viruses from spreading. Holding down the shift key during the startup of Word allows for a file to be opened without allowing any Auto macros to execute. This will prevent viruses that use the AutoOpen macro in order to spread. Similarly, if held down at exit, the AutoClose macro will not be executed.

Holding down the shift key during the startup of Word allows for a file to be opened without allowing any Auto macros to execute.

In order to correctly make use of this feature, one must be holding down either shift key at the moment Word is activated, and continue holding throughout the startup process. To be sure this occurs, you should hold the key down with one hand while the other hand is double-clicking the Word icon. It might seem obvious, but it is not always easy to do, even if you remember. The shift key must be held down for the duration of Word's startup process. Letting go early may allow a macro to execute.

PROS: Effective against the AutoOpen and AutoClose macros

CONS: Easy to forget or perform incorrectly
If done incorrectly, you will not know until it is too late
Viruses that do not use AutoOpen or AutoClose will still infect

DisableAutoMacros

DisableAutoMacros is a Word macro function does exactly what the name implies. If the function is activated, no auto functions will execute automatically until the function is turned off (or until the next Word session). Removing the ability to automatically execute the auto functions limits those viruses from easily infecting your system.

Ironically, the best way to invoke this function is through an AutoExec macro. However, in the following instructions, the end result will be an AutoExec function in its own template file, not in the normal.dot file. I recommend the template file be placed in the default startup directory in order to keep the normal.dot file pristine. And, in this manner, it is easier to give the file to others and harder for viruses to find and remove.

Start by making sure your *normal.dot* is writeable, and empty...

Click on **Tools, Macros...**
In the **Macro Name:** box, Enter **autoexec**
Click on **Create**.

Edit the macro to insert the DisableAutoMacros command:

```
Sub MAIN
DisableAutoMacros 1
End Sub
```

Close the editing session. Exit and save all changes.

In the DOS environment:

copy \msoffice\templates\normal.dot \msoffice\winword\startup\noauto.dot
erase \msoffice\templates\normal.dot

PROS: Disables all Auto macros
Greatly reduces chance of infection

CONS: Not foolproof
Does not disable other intercepted macros, key shortcuts, etc.
Environment is no longer pristine. May lead others to believe the macro you have established is suspicious and cause technical support issues.

Prompt to Save Normal Template in noauto.dot

In creating noauto.dot in the above process, it is beneficial to also include the command which turns on the Prompt to Save Normal Template choice. This can be accomplished by using the following macro in place of the above:

```
Sub MAIN
DisableAutoMacros 1
ToolsOptionsSave .GlobalDotPrompt = 1
End Sub
```

Or for Word 97:

```
Public Sub MAIN()
  WordBasic.DisableAutoMacros 1
  WordBasic.ToolsOptionsSave GlobalDotPrompt:=1
End Sub
```

This insures that the option is set every time, should normal.dot be deleted, or something resets the option. This also allows the network administrator to distribute one file which enforces two of the ideas instead of just one.

Menu Choices within Word

Macros are separate from the text and are not seen unless one goes looking for them. The most common way to check for macros would be through Tools then Macro... Unfortunately, viruses can intercept menu items. And the most commonly intercepted function is... Tools/Macro. This makes it unwise in a suspect situation to use Tools/Macro to determine if macros exist in documents. In the Customized Tools/Macro section below, you will be shown how to create your own replacement to Tools/Macro.

It is safe, however, for you to view macros in document files through the use of the Organizer function. The organizer function can be achieved through either File/Templates/Organizer... or Format/Styles/Organizer... or by creating your own following the instructions in the Customized Tools/Macro section below.

To see if macros exist in a document file without being affected by them, exit and restart Word without opening any documents. If you suspect that the normal.dot file or the startup environment may be infected, you need to rename the file and rename all the document files in the startup directory to other than DOC or DOT so Word can be started in a pristine state. To ensure that no virus affects your viewing of a suspect file, one must ensure that Word is started in a pristine state by following the previous directions.

To see if macros exist in a document file without being affected by them, exit and restart Word without opening any documents.

Upon starting Word, get to the organizer via one of the previously mentioned methods. Choose the Macros tab. On the left half of the box is a button labeled Close File. Click that

button to change the label to Open File... Click on Open File... to get a Browse box. Choose the target file to investigate. (By default, Word lists the .dot files. If the file you wish to investigate is not so named, you should change the Files of Type: drop-down box to All Files.) The filename will be shown. If any macros exist in the file, they will be listed in the big box. If not, normal.dot will show up again with its macros. Practice on a file that you know has macros in it.

PROS: Safely determines if any macros exist in the target file

CONS: Difficult to tell whether macros contain a virus
Complex set of tasks

Customized Tools/Macro

Because default menu items are often targeted and intercepted by macro viruses, it is important to know how to create menu items which will have the same functionality as those which would be intercepted. Following are instructions to create an equivalent to Tools/Macros...

Make sure the normal.dot is writeable…

Click on **Tools, Customize...**
Choose the **Menus** tab
Under **Categories** click on **Tools**
Under **Commands** click on **ListMacros**
For **Position on menu:** choose (**At bottom**)
Click on **Rename**.

Close the editing session. Exit and save all changes.

In DOS, remember to make normal.dot read-only again.

Following this, you will have an additional choice under Tools to list the macros in your document.

PROS: Bypasses the need to use **Tools/Macro...**
Not subject to virus payloads tied to Tools/Macro

CONS: Works until viruses start to interept **Tools/ListMacros**
Requires creation of customized menus to regain use of lost menu features

Word 7.0a and Word97

In Word 7.0a (available for Windows 95) and in Word97, a new check box was dded to Tools/Options.../General. This check box allows the user to be alerted if any macros exist in a document which is about to be opened. If such a document is encountered, the user is given the choice of stopping, continuing unaltered, or continuing with the macros disabled.

If the user chooses to continue with the macros disabled, the file is opened in a read-only state and cannot be changed. If you do not normally make use of macros, it may be wise to have the features activated by default.

Although simple to activate, some of the usability touches added to this feature create exceptions that are not readily apparent. These conditions have previously been documented by Vesselin Bontchev and revolve around certain conditions where Microsoft would expect macros to appear. Thus if those macros turn out to be viral, the initial warning will not alert because the macros were expected to be there.

PROS: Generally effective. If there's a macro in the document, it tells you so.

CONS: Prevents editing of any document with normal healthy macros
Can be defeated by viruses that fall under Word's exceptions list

Read-Only Recommended for Normal.dot

Word also allows for its own enforcement of normal.dot as a read-only file. Therefore, even if the file is not read-only, Word can still open it as if it were.

The downside to using this option is that each time you start Word, it will ask you if you wish to open the file as read-only. This can become bothersome and lead users to turn it off quickly. Of the techniques described in this paper, this is the most bothersome as it will interfere with a message even in a clean environment.

To set up this option:

Start a Word session and explicitly open the normal.dot file. (\MsOffice\Templates)
Click on **Tools, Options…**
Choose the **Save** tab. See instructions Prompt to Save Normal Template section
At the lower left under **File-Sharing Options for *normal.dot*,** check **Read-Only Recommended**
Close the editing session. Exit and save all changes.

PROS: Allows flexibility for those who want their normal.dot to be read-only occasionally

CONS: Asserts messages to you each time you start Word

Password Protect Normal.dot

Yet another offering under the **Save** tab of **Tools, Options...**, adjacent to **Read-Only Recommended,** is **Write-Reservation Password** (For Word97: **Password to modify**). If this choice is invoked, each time Word is started, the user will be asked to enter the password or open the file as read-only.

The advantage of this is that only select people will be allowed to modify normal.dot, and only if that person knows beforehand that he/she wishes to change it.

The disadvantage is that only select people can clean up such an infection. And on those occasions when *normal.dot* is allowed to be infected, no warning is given that it has been infected.

To set up this option:

Start a Word session and explicitly open the *normal.dot* file. (*\MsOffice\Templates*)
Click on **Tools, Options...**
Choose the **Save** tab. The instructions are the same as above.
At the lower left under **File-Sharing Options for *normal.dot*,** type a password into the **Write-Reservation Password** box. You will then be asked to confirm the password.
Close the editing session. Exit and save all changes.

PROS: Allows flexibility for those who occasionally want their *normal.dot* to be read-only

CONS: Asserts messages to you each time you start Word
Only that same select few can clean up the global infection

Word97 Lock VB Project for Normal.dot

As a user of Word97, there is yet another method to make *normal.dot* a write-protected document. This feature prevents modules from being created, viewed, or copied into the Template Project. Macro viruses, which would be Visual Basic modules, would fall into that class.

However, font selections, AutoText, and other stylistic choices and settings do not. Thus a user could change certain default choices in *normal.dot* without having to overcome protections, and also not allow the standard macro virus to infect.

To set this up, make sure the *normal.dot* is writeable:

Start a Word97 session
Press **ALT-F11** to open the **VB-Editor**
Click on **View, Project Explorer** to activate
Mark ***Normal*** in the **Project Explorer**.
Click on **Tools, Normal Properties...**
Choose the **Protection** tab
Check **Lock project for viewing** and set a password
Click on **File, Save Normal**
Close the editing session

PROS: Virus cannot bypass this unless it happens to guess your password
Users needing to change Autotext and Toolbars won't be affected

Using RTF (Rich Text Format) Files

The previous suggestions are mostly concerned with how to detect viruses. This section addresses how to avoid sending out an infected document, and in so doing, possibly protect your organization from being infected.

There is a rumor that says **.RTF** files cannot have macro viruses. First, let us specifically define what that means:

When using the **File/SaveAs...** function, there is a box that allows you to choose what type file you wish to save this file as (**Save as type:**). When one chooses **Rich Text Format**, the document is saved without macros. However, this does not apply to embedded documents. Even though the top level may not contain macros, embedded documents may. But most people do not use embedded documents, and this is one reason why they should not. Abiding by the rule that if a file does not have macros, it cannot have macro viruses, this then becomes a rather useful way to distribute one's finished work.

But just because a file has a **.RTF** extension does not mean that it is a Rich Text Format file. First, any file can have any name you designate. But mostly, WM/CAP, one of the most prevalent Word macro viruses, actually takes over the **File/SaveAs...** operation specifically to fool a user who tries to take advantage of this fact.

In light of all that, I suggest the following. When finished with your work and about to distribute it:

SaveAs... the document as ***temp.doc*** (keep the same type as the original document).
SaveAs... this ***temp.doc*** file as ***final.rtf*** (change the type to **Rich Text Format**).

After completely exiting Word, notice that ***temp.doc*** is smaller than the original document. This is because we asked Word to save the file afresh. In doing so, the process cleans up the unused or deleted sectors from the OLE2 file and rewrites the document into a fresh new file. This results use as small a file as possible to house all the information in the document.

Following that, we saved the file again as a Rich Text Format file. So, from "the smallest file as possible," anything having to do with macros is removed. This results in an even smaller file, even if there are no macros.

So, given that ***final.rtf*** is smaller than ***temp.doc***, you can be reasonably assured that ***final.rtf*** will not be an infected document. (And you may want to replace your original document with ***temp.doc***, as the new one is smaller.)

PRO: For distribution, the smallest possible file is sent
Your friends are not infected by your documents
You and your company are not embarrassed
You get in the practice of distributing only RTF files, a good habit to have

CON: It is a few extra steps
May not be perfect

Methods Related to the Operating System

Read-Only Normal.dot

Probably the most effective free macro virus protection method is to change the attribute of normal.dot to read-only.

Probably the most effective free macro virus protection method is to change the attribute of normal.dot to read-only. As it is so easy to do and quite effective, it is also the most talked about method on the Internet.

DOS has the concept of attribute bits. The most commonly referenced are the System, Read-Only (RO), Hidden, and Archive bits. The specific attribute bit which interests us is the Read-Only bit. If the RO bit is set, normal DOS system calls will refuse to write or change the file. Thus, in theory, if the *normal.dot* file is RO, no virus will be able to change it.

As noted before, a virus generally wants to change the global environment. This generally causes the normal.dot to be rewritten. However, if the RO bit is set, when Word opens *normal.dot*, it recognizes and stores this fact. When Word exits, it remembers *normal.dot* was RO and refuses any attempt to change it.

Anyone who uses macros would be severely limited by this approach, however, as it would require constantly modifying the read-only status. A user of macros can still operate with a RO *normal.dot*, however, if he/she stores all files in the default startup directory. Macros would be handled in the way we described in the *noauto.dot* section above.

Second, and a very important note, is that one doesn't realize a virus is active until AFTER exiting Word. The significant technical note is to recognize that Word informs you of the attempt to write to *normal.dot* when it exits. So, throughout the time you are using Word, files will continue to be infected without warning. Only on exit do you realize that something bad has been happening throughout the day.

At that point, of course, you can immediately shift into "virus forensics" mode. Forensics needs as much historical information as it can muster. To facilitate this, we would want to make use of the Most Recently Used (MRU) list. By default, Word's MRU list is set to remember the last four files opened by the user. These are going to be the files of interest anyway. The files saved on that day are the ones to be chased down if any of them had been sent to anyone.

With a default of only four, you would most likely need to increase the length of the MRU list to the maximum number of files you might expect to use during the average day. To do so, choose **Tools** then **Options…** then **General**. Go to the **Recently Used File List** and increase the number to its maximum.

PROS: System level protection
Slows down viral spread
Will know of an infection by the end of the day
Viruses cannot circumvent this to infect normal.dot in the same session

CONS: If user must constantly update his macros, productivity would be hindered
Could require significant backtracking if virus is discovered
May create false sense of security

Word Viewer

One of the most common infection scenarios involves receiving a document through e-mail, double-clicking on it, and having Word automatically open the document. This is governed by one of two setups. It will be either the e-mail program itself being programmed to activate Word based on whatever criteria it uses, or it will be based on the e-mail program making use of the registry.

In this section, we will cover the steps necessary to change the default association of .DOC and .DOT files to Word. This affects such activities as double-clicking, drag-and-drop, and Explorer.

The title of this section is *Word Viewer* because instead of using Word to read .DOC files, another program is used instead, one which does not support macros. WordView or WordPad are such programs. WordPad supports no macros. WordView has restricted support of macros. This section covers the necessary steps to redirect the registry associations of .DOC and .DOT from Word to WordPad or WordView.

Find *WordPad.EXE* (or *WordView.EXE*) on your system
Note the full pathname for the file
Find *REGEDIT.EXE* or *REGEDT32.EXE* on your system and execute
Under **HKEY_CLASSES_ROOT**, locate **.doc** and **.dot**
Traverse the structure until you locate **Shell** then **command**
Change the associated command to the full pathname of *WordPad.EXE*

HKEY_CLASSES_ROOT
.doc
  Word.Document.6
Word.Document.6
  shell
open
  command
C:\Program Files\Windows NT\Accessories\WordPad.exe "%1"

NOTE: All the commands have to be changed: Open, New, etc.

PROS: Does not allow macros of any kind to execute

CONS: Limited word processing capabilities
Avoid macros, but doesn't tell you they are there
Some e-mail programs disregard the registry

If you do not have a viewer, one can be retrieved, free, from:
http://www.microsoft.com/msword/internet/viewer/viewer97/license.htm

Replace Normal.dot Every Time

In normal DOS usage, there is the regular suggestion to boot clean before running anti-virus programs. While there is no need to enforce a clean Word environment before running anti-virus programs, it is still worthwhile to know that one's environment is clean before each new day's use.

One way this is accomplished is to delete and replace one's *normal.dot* file each day. We use *autoexec.bat* to force this each time the machine is booted. Of course this effectively makes *normal.dot* a poor mechanism to hold changes. If changes are not moved to the archived copy, they are lost. This method also does not inform the user of an infection. The infection is simply destroyed. Since that's the normal course, I've added the read-only attribute to the file.

To set this up, do the following:

cd \msoffice\templates
md ***archive***
copy normal.dot ***archive\normal.goo***
(Goo is short for "good" and presumably won't conflict with any other extensions in use.)
attrib +r ***archive\normal.goo***

Add the following to autoexec.bat:

pushd \msoffice\templates
erase /f normal.dot
copy ***archive\normal.goo*** **normal.dot**
attrib +r normal.dot
popd (If you do not have pushd/popd, use full directory pathnames)
endbat (transition to empty *endbat.bat* file, see endbat.bat section below)

PROS: Boots Word clean every day

CONS: Hassle to change any macros
Doesn't tell you if anything happened

Check For Changes to Normal.dot

Another way to ensure a clean startup of Word, instead of forcing it to be replaced with a new version each day, is to check the present version against the known clean one which had been archived. This method differentiates from the previous one in its ability to alert the user of changes having occurred. To achieve this, we have to save a copy of the current (hopefully clean) *normal.dot* file.

The setup:

cd \msoffice\templates
md *archive*
copy normal.dot *archive\normal.goo*
attrib +r *archive\normal.goo*

Add the following to *autoexec.bat*:

diff \msoffice\templates*archivenormal.goo* \msoffice\templates\normal.dot > NUL**
(Diff is a program similar to fc (from DOS). Fc does not return the necessary errorlevel for use in this manner. Ask local gurus for a copy of diff, they're sure to have one.);
if errorlevel 1 goto changed
[continue]
goto end

:changed
echo normal.dot changed^G
The "^G" represents a CTRL-G, which is "hold down the Ctrl key, and press G." It causes a beep.
pause
:end
endbat

PROS: Informs of any change
Ensures clean *normal.dot* each day
Generally transparent

CONS: Not effective until next bootup
Requires expert to setup
Requires expert to understand use

Check the Startup Directory

In the *DisableAutoMacros* section above, after creating noauto.dot, it was suggested that the file be placed in Word's Startup directory. Any template file stored in this directory is automatically installed into Word's environment when Word starts up. That also makes this directory the target of viral attacks as any virus could add itself to an environment by dumping a template file in this directory. Therefore, it is important to keep an eye on the

directory to make sure the contents of the directory do not change.

To make sure no viruses are added to the directory, we need to store a listing of the directory from a known clean state. Then, each time the machine is started, we make use of autoexec.bat to check that the current contents of the directory is not different from the list which represents what it should be. The code needed to make this happen can be seen below.

To set up:

Start Word and locate the default Startup directory
Click on **Tools**
Go to **Options…**
Choose the **File Locations** tab
Look for the entry related to **Startup**
(If there are three dots in the directory name, double click on the entry to see the full directory name. Write down this directory name. Below, the code example uses *msoffice**Winword**Startup* as that directory.)
Cancel, etc. and exit from Word

In DOS, continue by executing the following instruction:

dir /b /a \msoffice\Winword\Startup > %TEMP%\startup.lst

This dir command creates the list which lists the current contents of the Startup directory and stores this list in your defined "temporary" directory. If you have an older version of DOS, it may not have some of the parameters that are used.

To explain the instruction:

/b creates the short form of this command
/a includes hidden files so virus writers cannot use that to hide
%TEMP% is replaced by DOS with the defined "temp" directory
Add the following to autoexec.bat:
dir /b /a \msoffice\Winword\Startup > %TEMP%\startup.chk
diff %TEMP%\startup.lst %TEMP%\startup.chk > NUL
if errorlevel 1 goto diff_startup
[continue]
goto end
:diff_startup
echo Word startup directory changed^G
pause
:end
endbat

PROS: Informs of any change
Ensures clean boot up each day
Generally transparent

CONS: No warning until next boot up
Requires expert to setup
Requires expert to understand use

Using endbat.bat

In the battle against macro viruses, it is important to know that many macro viruses have payloads which attach extra code to the *autoexec.bat* file. When this happens, the next time the machine is started, the code added by the macro virus will execute. Thus, it is important to come up with a method which prevents such payloads from taking effect.

In the battle against macro viruses, it is important to know that many macro viruses have payloads which attach extra code to the *autoexec.bat* file.

With batch files, there are two different ways to transfer control to another batch file. One method is to "call" the second batch file. In this way, after the completion of the second batch file, control is returned to the "caller." The second method transfers control directly to another batch file. This method does not return control upon completion of the second batch file.

First, we create an empty batch file called *endbat.bat*. In the *autoexec.bat*, instead of letting it end by executing the last instruction, we transfer control to endbat, which finishes the startup process. With this setup, any code that is added by a macro virus to the end of *autoexec.bat* never gains control. Thus, none of that code runs. This same setup will cause software installations which add to the end of *autoexec.bat* to also fail in the same way. In such situations, simply remove the endbat transfer and replace it at the end of *autoexec.bat* again.

PROS: Endbat.bat immunizes against the effect of viruses

CONS: May interfere with software installations that write to autoexec.bat

Rename debug.com and debug.exe

Another favored method by virus writers to deposit payloads onto users' machines is by way of a debug script. (A debug script is readable text instructions sent to the DOS utility named **debug** to create a binary file.) Usually, this is used to deliver virus programs or other binary data. To combat this, one can rename or remove **debug** from users that don't normally have need for that program.

Verify that the program is no longer available by typing "**debug**" on a command line. If the program still runs, the job is not yet complete.

PROS: Eliminates effect of virus payloads that use debug to plant such onto machines

CONS: Not a solution for programmers who need to use debug

Rename WScript.exe

This next technique is similar to the previous section, involving a file named *WScript.exe*.

Virus writers have started to take advantage of VisualBasic Script, which requires the installation of something called the Windows Scripting Host. If your operating system is Windows 95 or WindowsNT, in order to have Windows Scripting Host installed, you must install it as an adjunct. But since Windows 98, the Windows Scripting Host comes installed by default. So, similar to renaming ***debug.exe***, you can rename or remove ***WScript.exe***.

PRO: Users not affected by viruses that use VB Script to invoke payloads or to spread

CON: Some people actually do want VB Script

Excel Macro Viruses

Check the XLSTART Directory

Excel has a startup directory similar to that of Word. Any template file found in this directory is automatically loaded into Excel on startup. This directory is the XLSTART subdirectory under where Excel exists. This behavior being exactly the same as Word, meaning we can use similar code as was discussed in the *Check the Startup Directory* section above.

dir /b /a *msoffice\Excel\XLStart* > *%TEMP%\xlstart.lst*

Add the following to autoexec.bat:

```
dir /b /a \msoffice\Excel\XLStart > %TEMP%\xlstart.chk
diff %TEMP%\xlstart.lst %TEMP%\xlstart.chk > NUL
if errorlevel 1 goto diff_xlstart
[continue]
goto end
:diff_xlstart
echo Excel startup directory changed^G
pause
:end
endbat
```

PROS: Informs of any change
Ensures clean boot up each day
Generally transparent

CONS: No warning until next boot up

Create a Personal.xls File

This technique is equivalent to creating a Payload macro to address Word macro viruses. Laroux.A checks for the existence of a file by the name of *personal.xls* in Excel's *XLSTART* directory. If one exists, it does not infect. Thus, if we put a file by that name in that directory, we will be immunized from Laroux.A. As with Concept.A, Laroux.A is the most widespread of all Excel macro viruses.

To create such a file, simply take an empty Excel file and place it in the *XLSTART* subdirectory under where Excel is located.

As in the case of Word, many other viruses have since appeared, and other variants of Laroux now exist that would not be affected by this approach.

PROS: Works against Laroux.A

CONS: Ineffective against any other virus

Excel Macro Virus Protection

Like Word97, Excel97 also has the **Macro virus protection** option as a check box available under **Tools/Options…/General** (see also the section entitled "Word 7.0a and Word97"). This check box enables the user to be alerted if any macros exist in the spreadsheet about to be opened. If such a spreadsheet is encountered, the user is given the choice of stopping, disregarding the warning, or continuing with the macros disabled.

However, Excel users often use more macros than Word users. Therefore, Excel users are more likely to get the alert involving useful macros than their counterparts who use Word. Thus, though users should be encouraged to use this option, system administrators would likely find rebellious users if they tried to enforce this technique.

PRO: Generally effective. If there's a macro in the spreadsheet, it tells you so.

CON: Many Excel spreadsheets have useful macros, so many false alarms

PowerPoint Macro Viruses

It took about two years after the introduction of PowerPoint97 for the first PowerPoint virus to appear. Because PowerPoint is much less widely used than Word or Excel, and because PowerPoint97 shares the same macro programming language as its Office97 brethren, the first PowerPoint97 viruses happen to be cross-infectors which can also infect Word or Excel.

PowerPoint only started to support macros in the version distributed as PowerPoint97.

PowerPoint only started to support macros in the version distributed as PowerPoint97.

Versions prior did not support macros, thus were not capable of having viruses. However, later versions, as with PowerPoint 2000, will also support macros and are therefore equally susceptible to viruses as the PowerPoint97 version.

Blank Presentation.pot

As with Word, PowerPoint has a default template mechanism similar to *normal.dot*. For PowerPoint97, this file is called Blank Presentation.pot and can be found in the …*Office**Templates* directory (*Program Files**Microsoft Office**Templates* typically). As it is a singular file, the techniques described in points entitles " ReadOnly *Normal.dot*," "Replace *Normal.dot* Every Time," and "Check For Changes to *Normal.dot*" are all equally applicable to PowerPoint. All the **PROS** and **CONS** stated for those sections apply as well.

PowerPoint Macro Virus Protection

As with Word and Excel in Office97, PowerPoint97 also has the **Macro virus protection** option as a check box available under **Tools/Options…/General**. (The option is not available for Access, the last of the four components of Office97.)

The technique works the same. Thus all the **PROS** and **CONS** for points entitled "Word 7.0 and Word97" and "Excel Macro Virus Protection" apply to PowerPoint97 as well. The only difference is that most users, while doing elaborate presentations, do not make use of macros to accomplish those feats.

Office 2000 Viruses

Office2000 perpetuates many of the anti-virus issues already discussed in previous Microsoft Office versions in this paper. However, a few specific issues arise in the Office2000 environment in particular. Additional tips for securing against viruses for those situations are offered below.

Access 2000

Access 2000 will finally support **Macro virus protection** under **Tools/Options…/General**, like the other Office products. See also points entitled "Word 7.0 and Word97," "Excel Macro Virus Protection" and "PowerPoint Macro Virus Protection."

Word 2000

Word 2000 will check the registry for settings to decide whether it will allow macros to execute or to always open a document with macros disabled. It will check in two locations. One of the registry locations will be under the HKEY_LOCAL_MACHINE tree, which will require someone with admin rights in order to change it. Thus, if it is set, and users are not granted admin rights, users can be forced to do without macros.

Script Viruses

...the Visual Basic Script language can be used by macro viruses.

As introduced in the point entitled "Rename *WScript.exe*", the Visual Basic Script language can be used by macro viruses. There are other methods than renaming *WScript.exe* that can make your system less vulnerable to an automatic, or silent, infection.

Suppress the default association of .VBS

Normally, the **.VBS** extension is associated to *WScript.exe*. With this association, a "double-click" on such a file will execute the script. So, instead of allowing a double click to automatically run the script, you might change it to bring it up in an editor. Of course, you would want to leave at least one association to run the script, after you've read it. If you choose this technique, follow along with the instructions from the section entitled "Word Viewer" and change it as below, from:

HKEY_CLASSES_ROOT
.vbs
VBScript file
VBScript file
shell
open
command
C:\Program Files\Windows NT\WScript.exe "%1"

To:
C:\Program Files\Windows NT\Notepad.exe "%1"

If **REGEDIT** scares you, you can make these changes by using **WINDOWS EXPLORER:**

View
Options...
Choose the "**File types**" folder.
In the **Registered file types** window, choose **VBScript file**, then **Edit...**
An **edit file type** window is displayed.
In the **actions** window, choose open.
An **editing action for type** window is displayed.
In the **Application used to perform action,** change the old command to the new one.

PRO: Double clicking a file will not automatically execute .VBS files

CON: Double clicking a file will not automatically execute .VBS files

Additional Tips

Handling Suspect Documents

For the network administrator who must handle a corporation's suspect documents, here are some additional suggestions.

First be sure to use all the techniques in the sections above. If a file is suspect, create a clean environment by using the process outlined in the "DisableAutoMacros" section (one of the "Methods Provided in Microsoft Word"). Examine the file using **File/Templates/Organizer**, before opening any other files. If the suspect file does not have a **ToolsMacro** entry, use it to rename the macros to shortened names before examining the macros. If it does, create your own **ListMacros** menu option and use it instead of **ToolsMacro**.

Cleaning Infected Documents

There is also the rare occasion that a network administrator may need to clean an infected document immediately so the document can be used without delay. As anti-virus vendors, we recommend strongly *against* this activity. Please use this technique with utmost care, and only if there is no time to employ the assistance of experienced virus researchers. This is best done on a stand-alone machine. But if that can't be accomplished, be *very* careful.

After verifying that the virus does not have an EditCopy or EditCut macro, and there are no templates in the startup directory nor *normal.dot*, open the file while holding the shift key. (Or for the adventurous, place *noauto.dot* into the startup directory.)

Select entire Document
Edit/Copy to Clipboard
File/Exit from Word
Delete normal.dot (or rename it) and remove all files from the startup directory
Restart Word
File/New a new empty document
Edit/Paste from Clipboard
File/SaveAs as a new document. In so doing, be sure that the file is not automatically being saved as a template. If so, the environment is infected. And assuming all the above was handled properly, pick up the phone and call your virus security vendor.

Author's Conclusions

Obviously, the most simple, effective, and reliable method of protecting against viruses is the use of proven anti-virus products. Because the purpose of this paper is to examine free techniques, however, I would like to conclude by noting which of the free methods I use personally.

Obviously, the most simple, effective, and reliable method of protecting against viruses is the use of proven anti-virus products.

To begin with, I set my *normal.dot* file to read-only. In addition, I use "Prompt to Save Normal Template." As detailed earlier, the two do not conflict and thus it's possible to use both. Both are meant to warn you by the end of the day if your environment was infected during the day.

But, why use both? Isn't one enough?

The first answer is that it doesn't hurt, so why not? The second answer is that some viruses try to undo one or the other. So, using both techniques means a virus has to attack both simultaneously to circumvent the protection. And if nothing is happening, both are transparent, so they will not disturb your everyday work.

I also use the "DisableAutoMacros template" as distributed in the separate file *noauto.dot*. Most viruses make use of an auto macro of one sort or another to spread. And all the viruses in the wild do. With this macro in place, viruses will not automatically activate and the chance of spreading something, even if you come in contact with it, is much smaller. Furthermore, as described in its section, an MIS director can create this file and send it to the whole company to be placed in the appropriate location. Thus this can have corporate wide impact with little effort.

Lastly, throughout all the Office97 products, each is programmed to alert if any macros exist in an incoming document, be that Word, Excel, or PowerPoint. The products in their default mode have the macro alert on. Don't turn it off until you hit your first false alarm. And even then, judge how much trouble the false alarm caused. If you feel that it was not a problem, leave the setting on. The alert is not perfect (the problem scenarios have been documented in Vesselin Bontchev's paper for the 1996 Virus Bulletin Conference). But until you experience a false alarm, it will not have caused you any issues.

Acknowledgments

Vesselin Bontchev, Anti-Virus Research, FRISK Software Intl.

Stefan Geisenheiner, Anti-Virus Research, Amsterdam, NL, Network Associates, Inc.

Raymond M. Glath, Sr., President, RG Software Systems

Jivko Koltchev, Anti-Virus Research, Santa Clara, CA, Network Associates, Inc.

Akihiko Muranaka, Tokyo, Japan, Network Associates, Inc.

Francois Paget, Anti-Virus Research, Paris, France, Network Associates, Inc.

Translations available in German, French, Spanish and Portuguese

Who's watching your network

For more information on products, services, and support,
contact your authorized Network Associates sales representative.

CORPORATE HEADQUARTERS

3965 Freedom Circle
Santa Clara, CA 95054-1203
Tel (408) 988-3832*
Fax (408) 970-9727

**Call for additional Worldwide Sales Offices*

Visit our Website

www.nai.com

6-TVD-FAV-001 9/99