

## Macro Virus Protection in the Microsoft Office Line, Part Two

Gabor Szappanos 2001-10-01

### Macro Virus Protection in the Microsoft Office Line, Part Two

by Gabor Szappanos

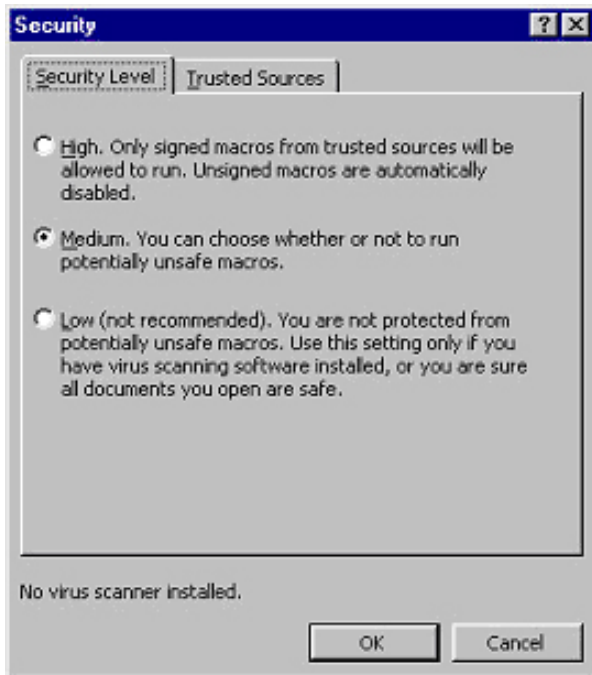
last updated September 26, 2001

---

The Microsoft Office programs are the most well known and widely-used programs in the world. They are also the most vulnerable targets for macro virus infection. One could easily blame Microsoft for not doing anything to prevent the virus threat; however, to do so would be to overlook the efforts that the software giant has made to diminish these threats. This is the second of a two-part series discussing some of the macro viruses that have targeted MS Office products. The [first article](#) looked at the macro viruses that affected earlier Microsoft Word and Office products. In this installment we will examine MS Office 2000, the new version of Microsoft Office, code named Office XP, and Outlook, Microsoft's e-mail program.

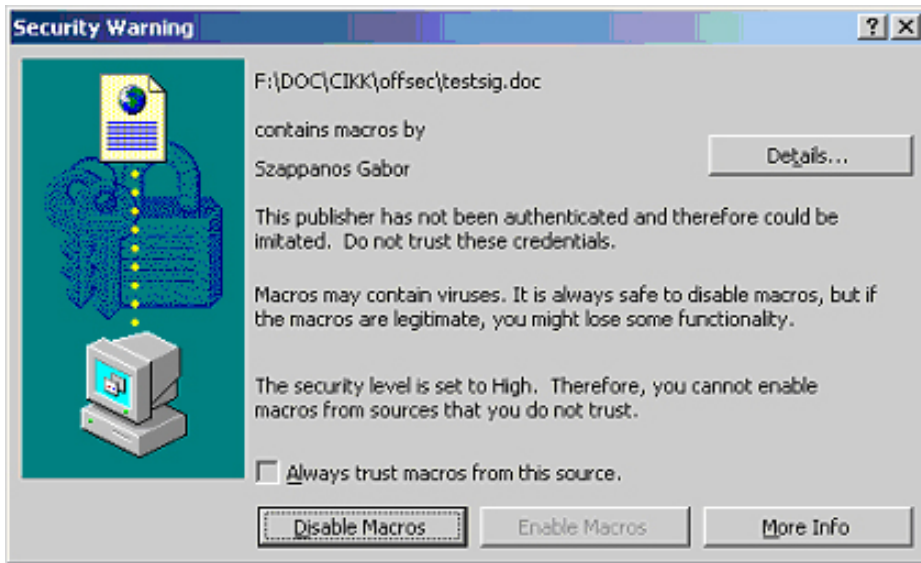
#### MS Office 2000

This family of products introduced a more granular macro security than ever before combined with the possibility to digitally sign macros. As indicated in the screenshot below, MS Office 2000 enables 3 security levels: high, medium and low.

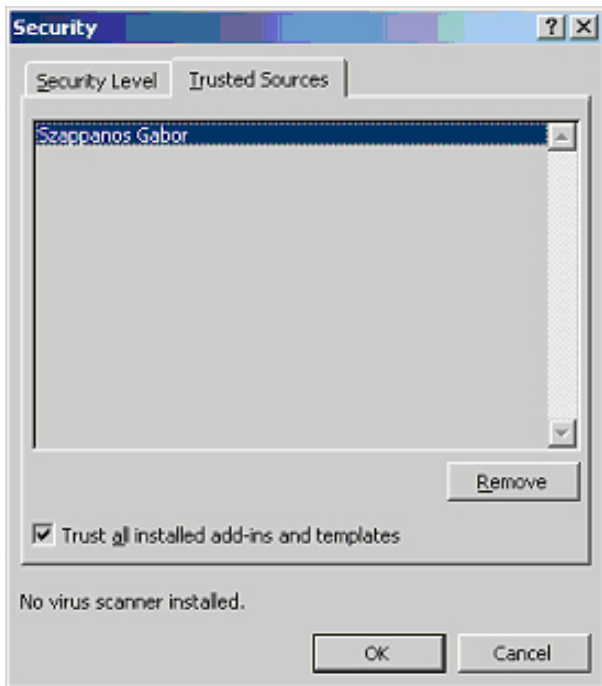


#### Windows 2000: High Security Level

The highest level allows the execution of those and only those macros that are digitally signed. If the signature is in the trusted list, the macro will automatically execute, otherwise the user is prompted, as follows:

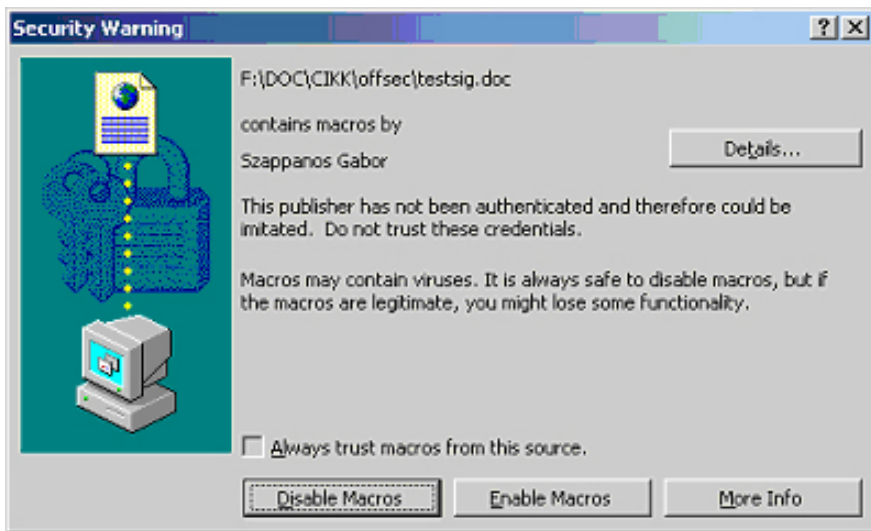


The user can either add the publisher of the macro to the trusted list by clicking the 'Always trust macros from this source' box or open the document with the macros disabled. If former feature is enabled, each macro signed with the same key will subsequently be trusted and executed without warning. Later on, the publisher can be removed from the trusted list. Unsigned macros are automatically disabled without any warning.



### Windows 2000 Medium Security Level

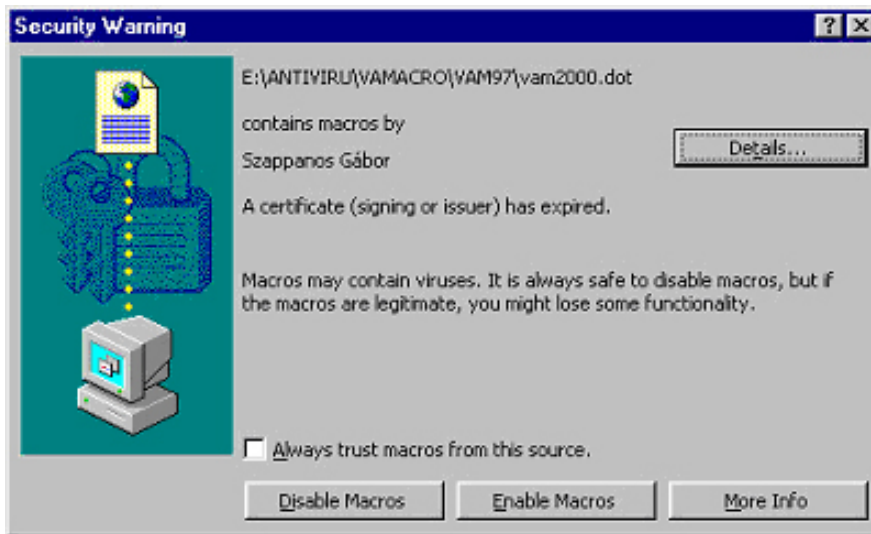
In the medium security setting, the user is prompted whenever a document containing macros is being opened. The exception to this rule is when the VBA document is signed with a digital signature that is on the trusted list, in which case the macro is executed automatically. However, as is illustrated in the screenshot below, when a document from an unauthorized or unrecognized source is being opened, a prompt appears offering the user the option of opening the document with macros enabled or, as is recommended, disabled.



## Windows 2000 Low Security Level

At the lowest security setting, all macros are executed, regardless of the digital signatures in them. Prior to the incorporation of digital signatures organizations were faced with a paradox: allow the execution of macros (thus leaving the system wide open for macro viruses) or disable macros (thus negating the benefit of using productivity-enhancing add-ons for Office products).

Now, Office users can sign their released products and the users can run only those macros that come from trusted source, thereby solving the old problem. However, even this solution could present users with problems. The user has to trust all products coming from the same developer and signed with the same key, while they would rather trust only selected programs that they use. However, while a macro virus infect a document that contains trusted macros, the checksum stored when the project was signed would not match the current state and Word will refuse to execute the macros.



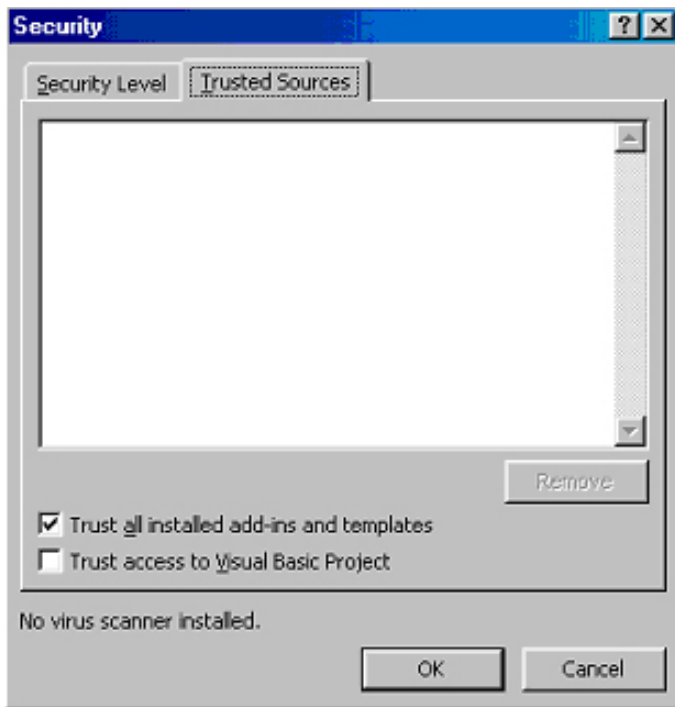
In case it is requested, detailed information is available about the digital certificate.



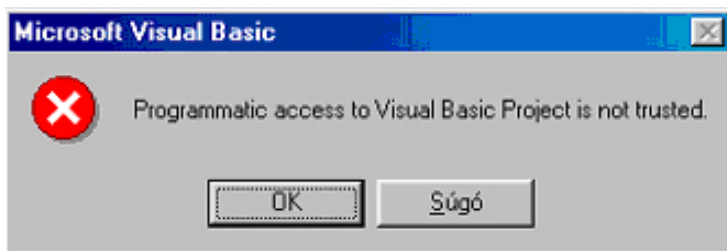
Unfortunately, as usual, the virus protection settings are stored in the Windows registry. It is more than easy for an external program, like virus dropper, to disable the security by settings it to the lowest level so that all macro viruses can activate.

### Office XP virus protection

As usual, the new version of Microsoft Office, code-named Office XP, brought new challenges to the virus experts. This time I am not speaking about the new file formats introduced in Access and Visio - this is quite normal and usual. What I am referring ot is one seemingly unremarkable little check box at the bottom of the Security dialog with the modest text "Trust access to Visual Basic Project".



As modest as it is, it represents a very important virus protection measure. When it is switched off (and it is the default installation option) and one attempts to open an older macro virus, it will fail to execute without any visible sign or error message. Investigating further and trying to open the same virus in the Visual Basic editor, the run attempt will pop up a run-time error message:



So what exactly is going on? The operability of practically all VBA macro viruses depends on the access to 2 adjacent object models: the application object model, which defines the event required for the activation of the virus (like the Document\_Open event that is fired whenever a document is opened,) and the VBE object model, which provides programmatic access for manipulation of Visual Basic code within the macro storage.

For example: the dots in the expression `Application.ActiveDocument.VBProject.VBComponents` (a commonly used expression in macro viruses) are not at all equal. The first and the third are references to the property of the object on the left side of the dot. The second is different: it is really a gateway between the two separate object models. It seems that Microsoft tried to seal the gateway between the two object models with this setting.

As most of the known macro viruses use the methods of the CodeModule object to proliferate (`InsertLines`, `AddFromFile`), and since the CodeModule is only accessible through the VBProject object, it would seem that this improvement could actually put a stop to the proliferation of all known and future macro viruses. However this option can be turned on and off.

It is a shame that Microsoft could not make up its mind and completely remove access to the VBE object model. There are some applications, such as code management tools like Code Librarian in Microsoft Office 2000 Developer or self-modifying VBA solutions that required access to the VB project objects, and this was enough reason to leave it in. As it is a selectable option, the access to the VBA project can be granted.

This is not a big problem: users are not likely to turn it on, as the isolation of the VBE object model will not be noticeable to 99.9% of them. The only problem is that the value of this setting is stored in an obvious location and under an obvious name in the registry, which makes it extremely easy for a virus to disable it and go on with the infection. Office applications read this setting during startup, and do not notice any change in it until the next startup, so the infection procedure has to happen in two sessions: in the first session, the access is granted, and in the second session the actual intrusion takes place. This has already been done. Even before the final version of Office XP was released, there were already 2 viruses, Listi.A and Pcut.A that are perfectly capable of working effectively in Office XP.

So how much protection will this security "enhancement" provide? Even less than the improvements introduced in Office 97 Service Release 1. It could effectively stop the migration of the vast majority of currently known macro viruses, but the Office XP aware macro viruses that will without doubt appear will easily bypass it. This does not mean however, that the upconverts of the Office 97 viruses will not appear. As this protection can be turned off, there will always be users, who will do that, thus successfully upconverting non-Office XP aware macro viruses. Consequently, the upconversion issues should be addressed anyway.

## **Outlook Virus Protection**

The most popular target of current e-mail worms is the Outlook e-mail client, which has been an integrated part of the Office product suite since Outlook 97. But the first version did not provide enough support for viruses, as it supported only limited Vbscript programming capabilities. Outlook 98, and all subsequent versions featured the VBA programming environment. As a side effect, the Outlook object model is accessible through Activex automation. Using this, any external program, even a simple VBScript can access Outlook and, by reading the address book, and can make Outlook send infected messages to all selected recipients.

This was not problematic until the Melissa and Loveletter viruses appeared. These viruses used the Outlook ActiveX programmability interface to drive Outlook to send massive number of copies of itself. It was at that point that Microsoft reacted again and released a security patch for Outlook 2000. The features introduced in this patch lived on in a somewhat more manageable way in Outlook XP.

This patch introduced two security additions. The first restricts the possible file extensions that can be sent and/or received by e-mail. The following screenshot represents an extensive list of possibly dangerous file extensions that are blocked by Outlook.

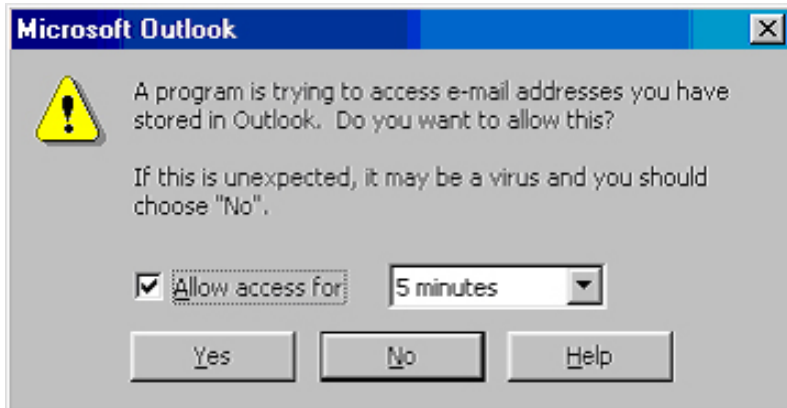
File extension	File type
.ade	Microsoft Access project extension
.adp	Microsoft Access project
.asx	Windows Media Audio / Video
.bas	Microsoft Visual Basic class module
.bat	Batch file
.chm	Compiled HTML Help file
.cmd	Microsoft Windows NT Command script
.com	Microsoft MS-DOS program
.cpl	Control Panel extension
.crt	Security certificate
.exe	Program
.hlp	Help file
.hta	HTML program
.inf	Setup Information
.ins	Internet Naming Service
.isp	Internet Communication settings
.js	JScript file
.jse	Jscript Encoded Script file
.lnk	Shortcut
.mdb	Microsoft Access program
.mde	Microsoft Access MDE database
.msc	Microsoft Common Console document
.msi	Microsoft Windows Installer package
.msp	Microsoft Windows Installer patch
.mst	Microsoft Windows Installer transform; Microsoft Visual Test source
file	
.pcd	Photo CD image; Microsoft Visual compiled script
.pif	Shortcut to MS-DOS program
.prf	Microsoft Outlook profile settings
.reg	Registration entries
.scf	Windows Explorer command
.scr	Screen saver
.sct	Windows Script Component
.shb	Shell Scrap object
.shs	Shell Scrap object
.url	Internet shortcut
.vb	VBScript file
.vbe	VBScript Encoded script file
.vbs	VBScript file
.wsc	Windows Script Component
.wsf	Windows Script file
.wsh	Windows Script Host Settings file

If an incoming mail contains a dangerous attachment, Outlook will remove it, after which it is not possible to retrieve the attachment. If an outgoing mail contains a file from the exclusion list, the user is warned that the recipient may not receive the attached file, but the e-mail goes out anyway.

This exclusion list is stored in the registry, and although Microsoft claims that an ordinary user would not be able

to modify it, there are already utilities that enable the modification of this list with a couple of mouse clicks.

The other addition that the patch included is the Object Model Guard, which drops an alert whenever an external program attempts to access the Outlook address book.



When this happens, the user can grant access to the external program for a couple of minutes. This may come handy in case one runs a mail delivery utility. As most of the known e-mail worms use the Outlook address book to pick recipients, this block can seriously improve the security. However it could be annoying in a real-life Loveletter incident, when this alert box comes up several hundred times and the user will have to deny access each time.

## Conclusion

This concludes our brief overview of macro viruses and Microsoft's Office Suite. As we have seen, Microsoft has taken some admirable steps towards addressing the very serious issue of macro virus protection in their Office Suite over the years. That having been said, there are still some measures to be taken to ensure that malicious users are precluded from disabling protective measures. Furthermore, while Microsoft is often expected to provide protection against macro viruses unilaterally, it would be helpful to remember that in the end, more often than not, it is user error that facilitates the successful proliferation of viruses. So while it is justified to expect Microsoft to take the necessary steps to ensure the quality of their software, users must take responsibility for their own role in virus protection as well.

[Privacy Statement](#)

Copyright 2006, SecurityFocus