Windows 9X in a Bad Neighborhood

Terry Wehunt

Windows 9X in a Bad Neighborhood
Terry Wehunt
October 28, 2001

This paper discusses security of Windows 9X machines under the control of certain registry settings and the impact of malicious code [see Notes 1] on maintaining registry setting. Specifically, it discusses the Internet Explorer registry settings. In the Windows 98 Resource Kit, Microsoft authors state "Internet Explorer is an integrated suite of Internet software that includes a customizable browser built on open Internet standards. It delivers an Internet solution to network administrators, who can customize and control their users' Web-browsing capabilities and ensure the security of their corporate Intranets."[MT] This paper argues a contrary position in that Windows 9X machines, while relatively safe in isolated LAN environments, may now be inherently unsafe in the environments where Internet connectivity, enhanced email, and macro enhanced Office products are common. Specifically, this paper argues that network administrators can no longer ensure security of their corporate Intranets by way of customizing and controlling Internet Explorer on Windows 9X machines.

One of the guiding principles of security is that overall security will only be as strong as the weakest point. In terms of physical security, this weak point may be a poorly anchored doorframe, a weak lock, or a weak wall. In terms of Window 9X security, the weak point may be the registry. The registry is a hierarchical database or virtual warehouse of hardware and software information and preferences. Ron Petrusha states, "Quite simply, the registry is the glue that holds Windows together". [PR, p 13]

Nobody would locate a weakly constructed, impossible to secure warehouse full of valuable goods in a bad neighborhood known for burglaries. The new bad neighborhood is the Internet. 99.99+% of the things and people there will do us no harm. It is that less than .01% that will do us in. Perhaps Windows 9X machines are connected to the Internet out of necessity, the costs of them not being connected, or because we believe, with proper measures, we can get by with it.

In the "Information Security 2001 Industry Survey", Andy Briney reports, "Cyber attacks double over the last year."[BA, p 34] This survey shows that 90 per cent of the respondent organizations said that they had suffered infections despite the fact that 88 per cent had antivirus software. Also, 78 per cent stated that employees had installed unauthorized software.[BA, p 34 – 40]

In isolated LAN environments, it was relatively easy to gain a good level of security on Window 9X machines. By using Microsoft supplied tools or third-party tools, changes could be written to the Windows 9X registry to limit users and to limit hostile code. Microsoft supplied a number of excellent tools that allowed administrators to make changes on Windows 9X machines without having to resort to direct edits to the Registry. The System Policy Editor was used to make user specific and computer specific changes. The Internet Explorer Administration Kit allowed the administrator to customize Internet Explorer packages with custom security settings. [MP1]

The old neighborhood where Windows 9X was often found was a fairly safe place. The only users of concern were those who were given access to the LAN. Even with the limitations of lack of local file and folder security, as well as the lack of granularity of user rights and privileges,

Windows could be made fairly secure. Users activities could be controlled by use of tools such as the System Policy Editor, use of roving mandatory profiles, access to the CD-ROM drives and the floppy drives could be locked, access to administrative tools could be removed, and email was plain text with no attachment capabilities.[MP2]

In the old neighborhood, administrators locked the doors and most slept well at night without worry. Administrators slept well at night knowing that they had denied users the only two recognized means of directly editing the registry. Users were denied access to registry editing utilities and to unapproved code that might access the registry via the Win95 API.[PR, p 17]

In the original product offering, Windows 95 came without an integrated web browser. The early versions of Internet Explorer were easy to delete in environments where a browser was not needed or another browser was preferred. However, on September 30, 1997, Microsoft released Internet Explorer 4.0. Internet Explorer 4.0 completely integrated into Windows 95 and introduced Security Zones. With Security Zones came the ability to have granularity of control over active web content. With the enhancements to security in Internet Explorer 4.0 and the use of the Internet Explorer Administration Kit, it became possible for administrators to select what types of active content would be allowed based on a trust model. The move from Internet Explorer 3.X to 4.X may be viewed as revolutionary. The revolution centers on Microsoft's inclusion of Internet Explorer in every version of its operating systems and the use of Internet Explorer code sharing.[SS]

Internet Explorer 5.X and 6.X may be viewed as incremental improvements on the foundation of Internet Explorer 4.x. Internet Explorer 6.0 has made an advance in security by including "an unsafe file list that is coded in the Sddocvw.dll file". Examples of unsafe file extensions are .exe, .vbs, .js, and .bat. However, one may override the unsafe file list by "adding a key with the appropriate file extension" in the registry settings. In addition, Internet Explorer now disables Active Scripting by default in the Restricted Sites Zone.[MPPS1]

Let us stipulate what the typical Windows 9X machine that we might find today would be like. This typical machine would probably have some version of Microsoft Office loaded, some version of Internet Explorer 4.01 through 6.0, Windows Scripting Host, both VBScript and JScript engines are present, media playing software will be onboard, and there will be some form of anti-virus software. Chances are this machine has Internet access. Chances are it receives email with attachments. Our users may be more computer savvy that we found in the old stand alone LAN days and have greater expectations from their computer. They will range from being brilliant and cautious to the other end of the spectrum. The one thing they all have in common is they all have de facto administrative powers on their Windows 9X machines. Any code they execute, intentionally or otherwise, will run on the machine with the powers of an administrator.

With Windows 98 Microsoft included the Windows Script Host (WSH) (originally known as ActiveX Scripting and now know as Windows Scripting Host). Prior to the WSH, DOS batch file techniques were used for many simple administrative and day-to-day tasks. "WSH extends the … command shell and lets you create robust scripts that you can execute directly on the Windows desktop or command console without having to embed scripts in a HTML document…"[MC] Prior to WSH, a user wishing to write and run a script (most notably JavaScript

or VBScript) was forced to embed the script within a HTML document and run it in a browser. With WSH one could run a script on the desktop or from the run line.[MT]

The details of the architecture of WSH are available in a number of books and from papers at the Microsoft Developers Network site. More importantly to this paper is what WSH enables. WSH allows objects, including all components of the registry, to be exposed to scripting languages. Software packages are beginning to use the capabilities made available through WSH. As an example, the Windows Update service appears to be dependent on WSH being available. With WSH, an administrator can write simple scripts that automate simple and complex tasks.[SW]

Microsoft provides an excellent document, "Description of Internet Explorer Security Zones Registry Entries", outlining the Zones keys, values, and DWORD values. Using the information from this document and the powers of WSH, an administrator may write a script to modify Window 9X registry setting to enhance security. Regardless of which version of Windows 9X is loaded, all Internet Security zone settings are under the control of registry settings. The zone security settings are found in two Registry keys:
HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\Zones
And
HKEY_LOCAL_MACHINE\ SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\Zones (Only used when HKEY_LOCAL_MACHINE\ SOFTWARE\Policies\Microsoft\Windows\CurrentVersion\Internet Settings\Security_HKLM_Only has a DWORD value of 1. Else, the CURRENT_USER register key controls the zone settings)
Other Registry keys of interest due to their impact on security are:
HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\ZoneMap\Domains
And
HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\SafeSites
[MTI]

As a proof of concept, a short five-line script that allows Active Scripting and Unsigned ActiveX controls was written and tested. I wish to thank Scott with Microsoft Security Center[PC1] and Russ Cooper of NTBugtraq[PC2] for reviewing and commenting on this proof of concept .vbs/.wsh script that overwrites Internet Explorer Security Zones. However, further research revealed that the script is one component of the VBS.Merlin.B and VBS.Merlin.C worms.[HK1 & HK2]

The script was loaded on several Windows 9X machines and was executed. These machines included one ME machine with Windows Scripting Version 5.6, a Windows 95 machine, and several Windows 98 machines. Several attempts were made to limit the concept .vbs/.whs script from being able to run from the desktop while not removing all Visual Basic Script and WSH capabilities.

The results of changing KEY_CURRENT_USERS\Software\Microsoft\Windows\CurrentVersion\Internet Zones\Zones\0\1400\ value to "1" (actually, hex notation must be used in the script) failed to produce a warning when the script was executed. It was thought that an Internet Explorer warning might have resulted. As a check on this method, the registry settings were left unchanged and a .chm file was opened. The .chm file produced the warning shown in Figure 1.

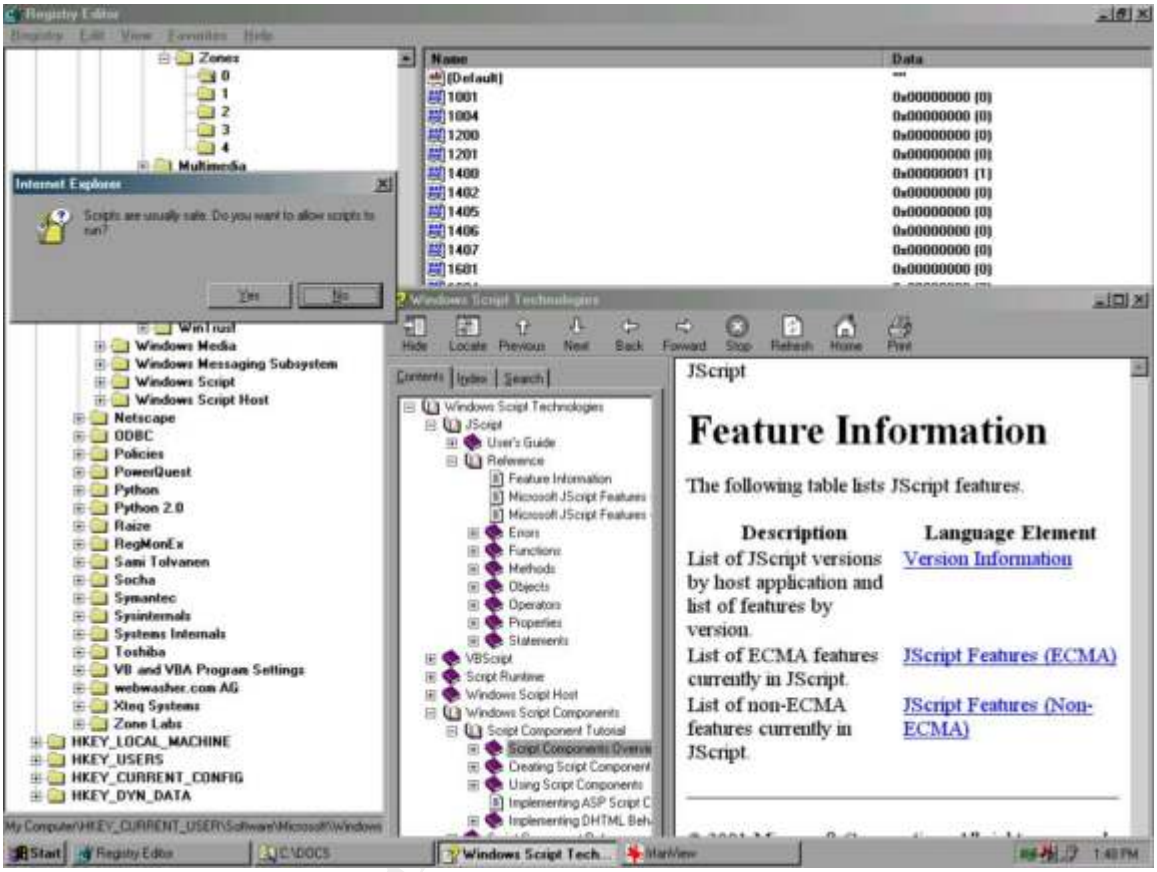


Figure 1. Internet Explorer Warning Box

The following were also attempted without success:

1. Editing HKEY_CURRENT_USER\Software\Microsoft\Windows Script Host\TrustPolicy to set the Value to 2. The value of 2 "means do not prompt the user and do not run untrusted scripts"[LE] or the value 2 means "prompt the user if the script isn't signed with a trusted certificate".[CA] Which one is correct is not important here, as no prompt was gained and the script ran successfully.
2. Editing HKEY_CURRENT_USERS\Software\Microsoft\Windows\CurrentVersion\Internet Zones\Zones\0\ . 0 designates the My Machine Zone. DWORD values for Values 1001 through 1805 were set to 1. A 1 setting normally produces a warning.
3. HKEY_LOCAL_Machine\Software\Microsoft\Windows\CurrentVersion\Internet Zones\Zones\0\. DWORD values for Values 1001 through 1805 were set to 1. A 1 setting normally produces a warning.

4. Editing HKEY_CURRENT_USERS\Software\Microsoft\Windows\CurrentVersion\Internet Zones\Zones\0\ . 0 designates the My Machine Zone. DWORD values for Values 1001 through 1805 were set to 1. A 2 setting normally disables an action.
5. HKEY_LOCAL_Machine\Software\Microsoft\Windows\CurrentVersion\Internet Zones\Zones\0\. DWORD values for Values 1001 through 1805 were set to 2. A 2 setting normally disables an action.
6. Norton AntiVirus 2001, running auto-detect, with the heuristic level set at the recommended level, with current signatures, and script blocking enabled.

Not only did Norton AntiVirus fail to recognize the concept script as potentially hostile; we can disable portions of Norton's functionality by rewriting registry setting with a script. Depending on the version of Norton AntiVirus the settings may be found at KEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\naavpsvc or elsewhere in the registry.[ST]

A second test of the concept script involved Microsoft Word. In 1996, Visual Basic for Applications (VBA) was included with Word as a replacement for Word Basic. In 1997, VBA 5.0 (was) launched, covering the complete Office 97 range of products. With VBA 6.0, launched in 1999, came a macro virus scanner API and the ability to choice three levels of macro security. The default level for most applications warns the user that a macro is present and allows the end user to choose whether the macro may execute.[SJ] An example of a security warning dialog box is shown in Figure 2

.

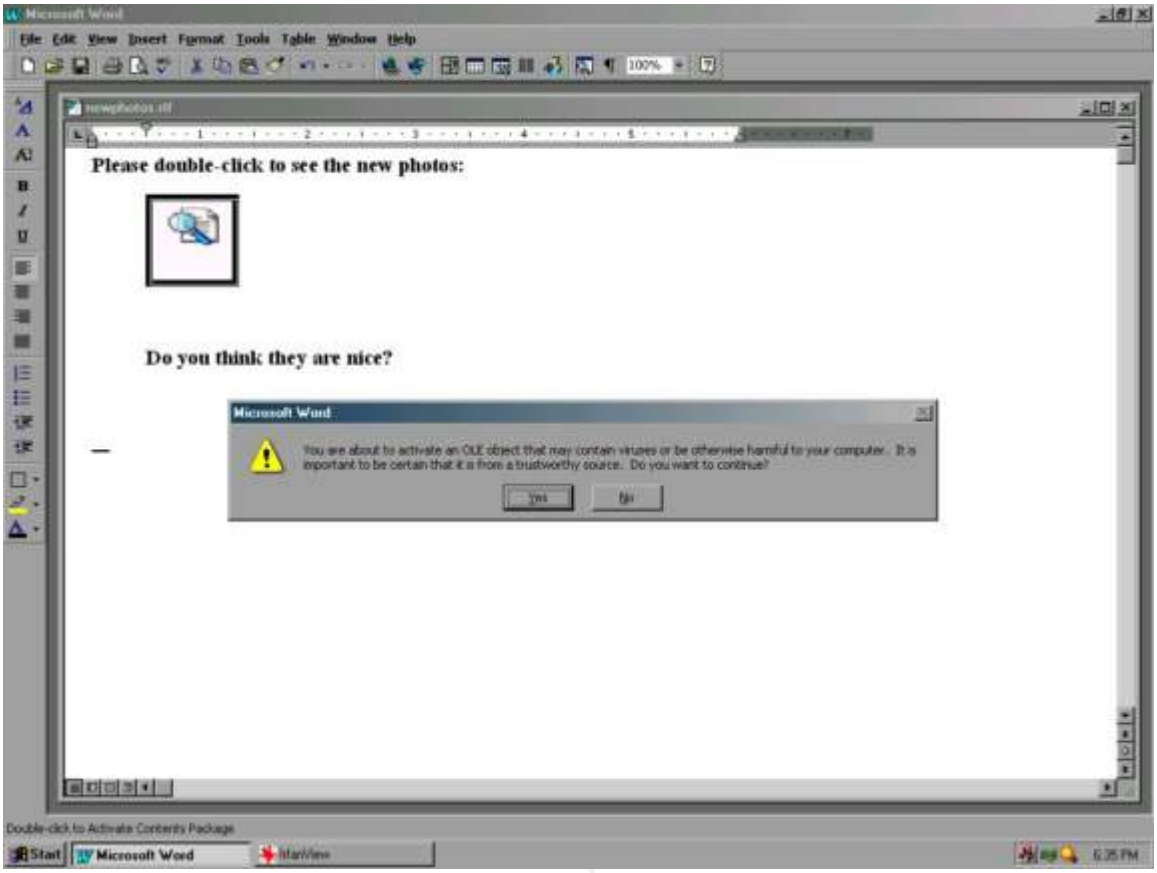


Figure 2. A Microsoft Word Security Warning Dialog Box.

This screen capture contains a packaged .vbs/.whs script (the concept script) that rewrites Internet security zone settings. At the time the script was packaged the normal .vbs icon was changed to help obscure the fact that it is a script. This was easily accomplished with Windows Word 97 SR-1 Professional.

Martin C. Carlisle and Scott D. Struder's work "Reinforcing Dialog-Based Security" shows how easily security dialog boxes may be suppressed and automatically answered using VBScript interfacing WSH. While Carlisle and Struder's paper is focused on the Outlook 2000 SR-1 E-mail Security update, it is an important paper in that it brings most forms of Microsoft based dialog boxes into further question. Using the code and discovery methods from the Carlisle and Struder paper, new code was written. Running this code in conjunction with the concept script, it was possible to suppress and answer the Microsoft Word Security Dialog box shown above. The script executed without warning and wrote to the registry.[CM]

Screen captures of the registry settings before and after execution of the concept script were made. Figure 3 shows Internet zone settings that were tightened to prevent certain active content from running on the machine.

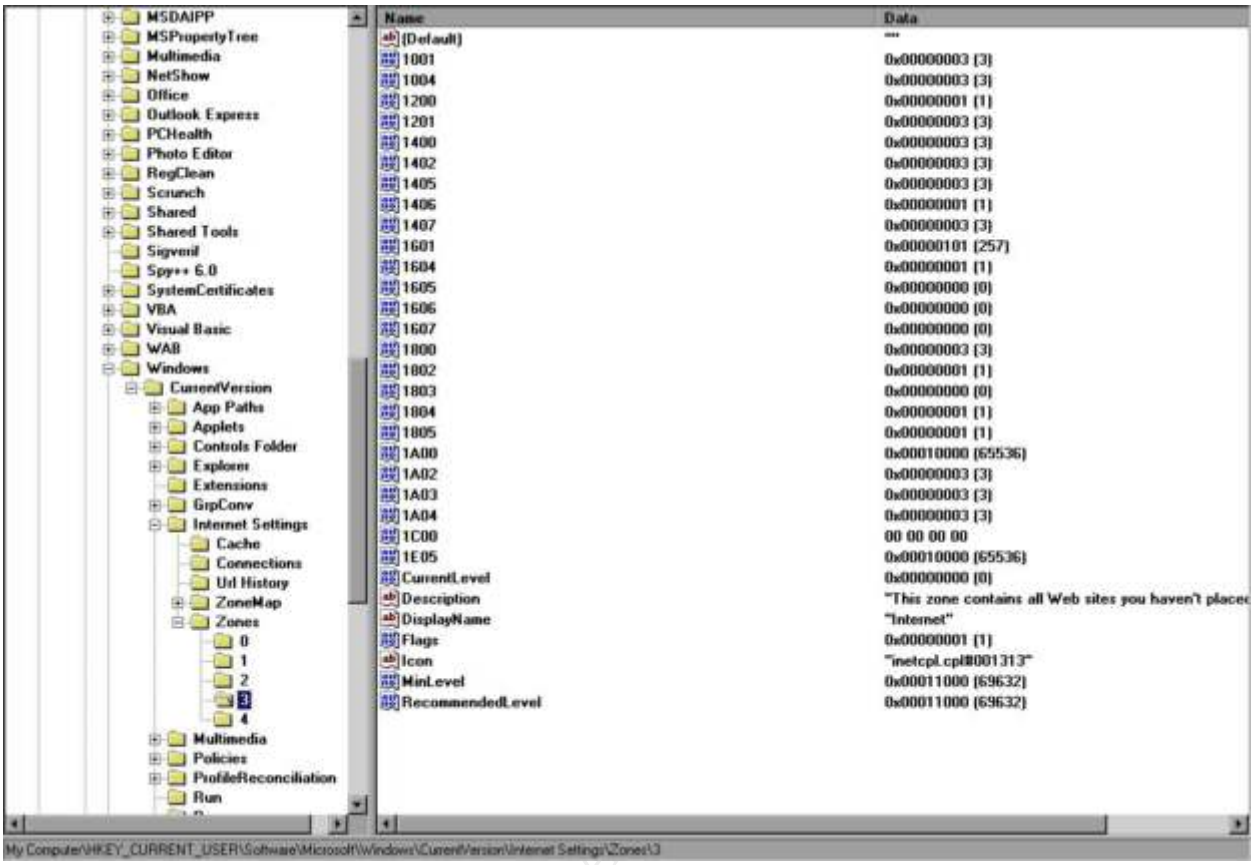


Figure 3. Internet Explorer Registry Settings before Execution of the Concept Script.

Then after the script ran, the Internet zone setting had been altered as shown below in Figure 4.

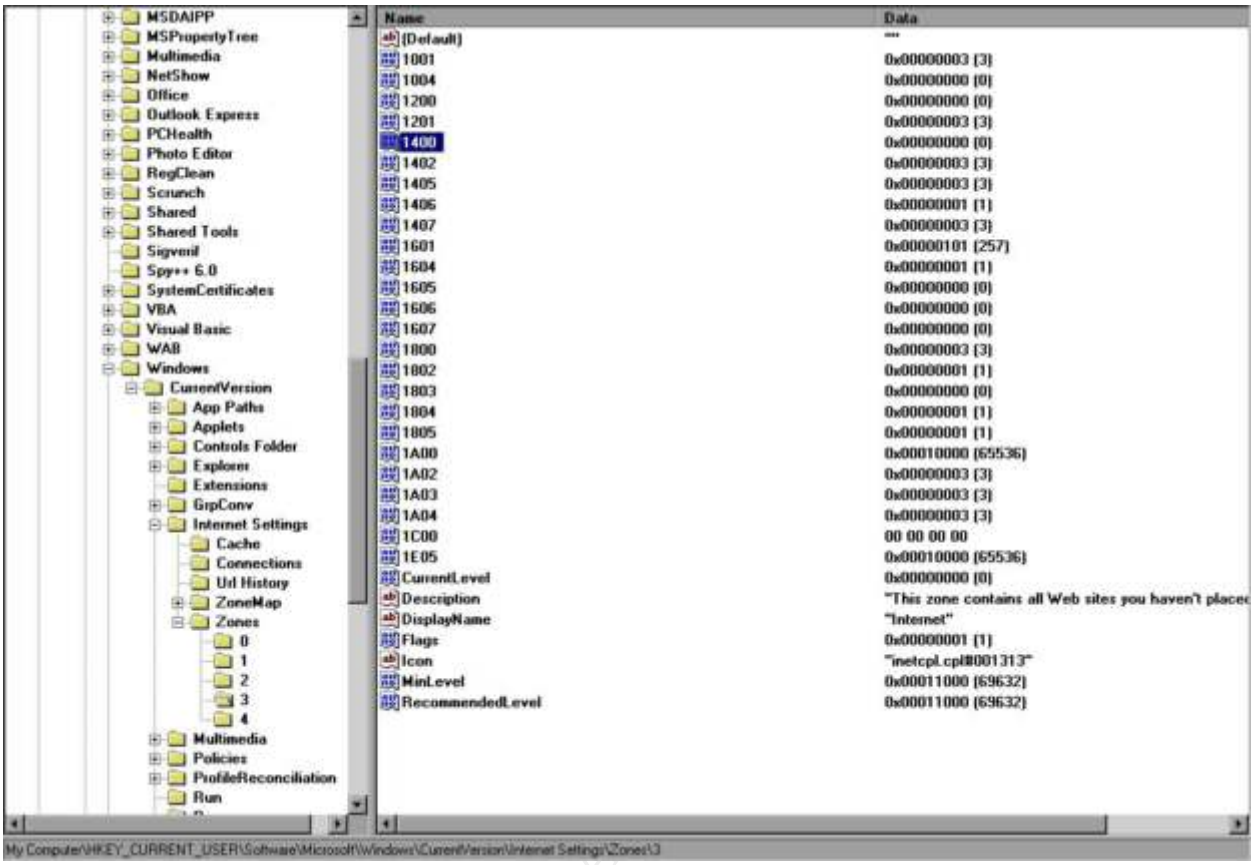


Figure 4. Internet Explorer Registry Settings after Execution of the Concept Script.

The changes written to the registry were to the DWORD values for Values 1004, 1200, and 1400 in the Internet zone. These changes would, in part, allow unsigned ActiveX controls to download and execute from any site not contained in the Local Intranet, Trusted sites, or Restricted sites.

The concept script shows that certain registry settings related to security may be changed on a Windows 9X machine. Once weakened, the machine might be pointed to an Internet site where an unsigned malicious ActiveX control would be downloaded. In the machines weakened state, the malicious code might load without being detected.

Many of the changes made to the tighten security of Windows 9X machines are made by changes in the registry. The following figure (Figure 5) is the Macro Virus Protection setting for PowerPoint 2000. The '1' setting shows that protection is on. The concept script could have been written to change this value as well.

.

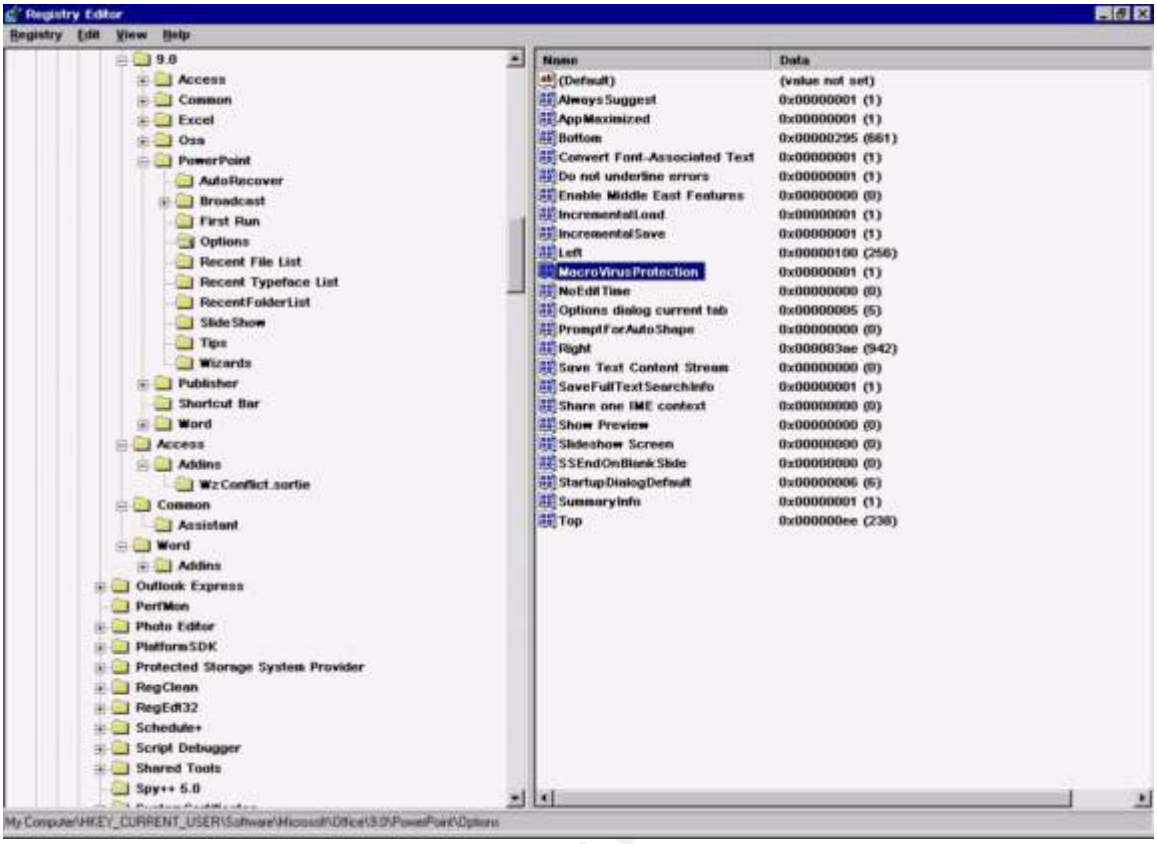
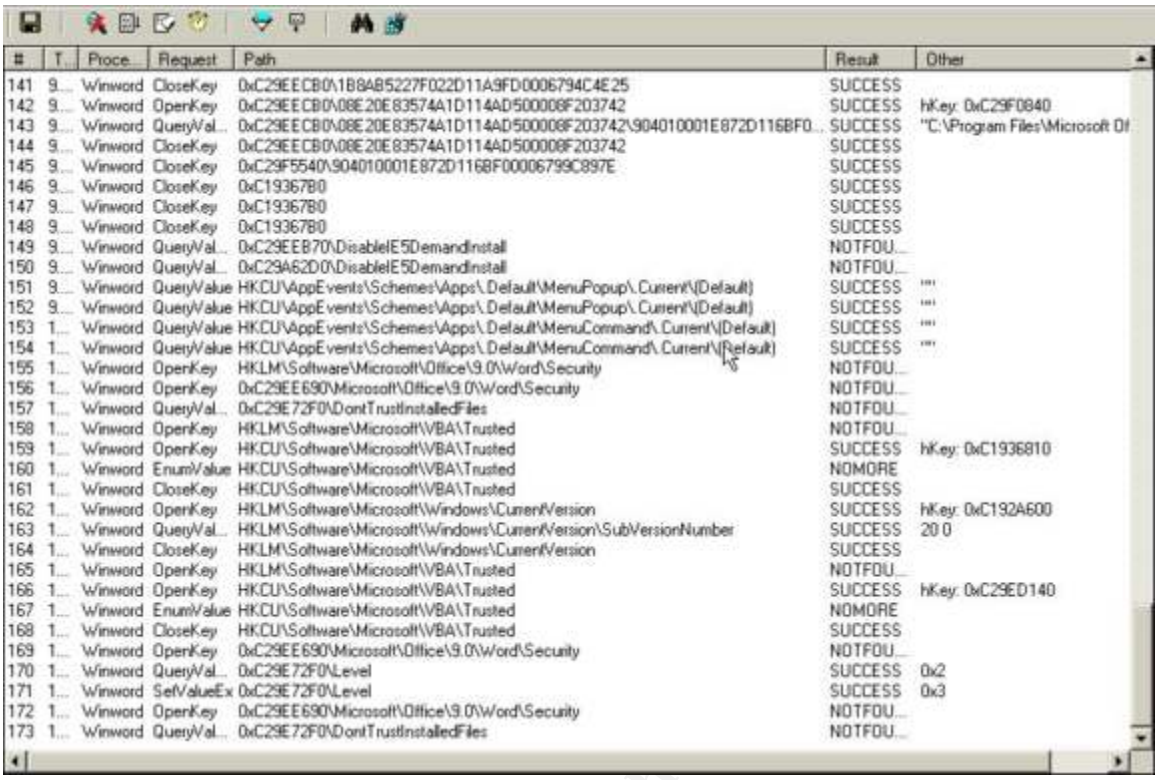


Figure 5. Registry Setting for PowerPoint Macro Virus Protection.

Outlook 2000 SR-1 security warnings can be eliminated on a file type by field type basis by adding a String Value of RemoveWarningFileTypes to HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft|Office\9.0\Outlook\Security. One simply adds the file extensions under the String Value to eliminate warnings.[MPPS2] Outlook XP continues the tradition of allowing registry edits to diminish security.[SG]

Microsoft has made a change in Office XP that is a move in the right direction. By default, access to Visual Basic Project is blocked until enabled via a checkbox. By blocking access to Visual Basic Project, many older macro viruses will be blocked. Unfortunately, the registry setting for this feature is easy to change programmatically.[SG]

While outside the main purpose of this paper to list all the registry values related to Window 9X security; it is easy to discover the values. Using tools such as Registry Monitor from Sysinternals one may monitor the registry while making changes to the security settings options within an application (Figure 6). By monitoring the registry in this manner one may better understand which registry values relate to the security settings for any application. For example, we can make changes to the security settings for Word and monitor the effects on the registry.



#	T...	Proce...	Request	Path	Result	Other
141	9...	Winword	CloseKey	0xC29EECB0\1B8AB5227F022D11A9FD0006794C4E25	SUCCESS	
142	9...	Winword	OpenKey	0xC29EECB0\08E20E83574A1D114AD500008F203742	SUCCESS	hKey: 0xC29F0840
143	9...	Winword	QueryVal...	0xC29EECB0\08E20E83574A1D114AD500008F203742\904010001E872D116BF0...	SUCCESS	"C:\Program Files\Microsoft Of
144	9...	Winword	CloseKey	0xC29EECB0\08E20E83574A1D114AD500008F203742	SUCCESS	
145	9...	Winword	CloseKey	0xC29F5540\904010001E872D116BF00006799C897E	SUCCESS	
146	9...	Winword	CloseKey	0xC19367B0	SUCCESS	
147	9...	Winword	CloseKey	0xC19367B0	SUCCESS	
148	9...	Winword	CloseKey	0xC19367B0	SUCCESS	
149	9...	Winword	QueryVal...	0xC29EEB70\DisableIE5DemandInstall	NOTFOU...	
150	9...	Winword	QueryVal...	0xC29A62D0\DisableIE5DemandInstall	NOTFOU...	
151	9...	Winword	QueryValue	HKCU\AppEvents\Schemes\Apps\.Default\MenuPopup\.Current\(Default)	SUCCESS	""
152	9...	Winword	QueryValue	HKCU\AppEvents\Schemes\Apps\.Default\MenuPopup\.Current\(Default)	SUCCESS	""
153	1...	Winword	QueryValue	HKCU\AppEvents\Schemes\Apps\.Default\MenuCommand\.Current\(Default)	SUCCESS	""
154	1...	Winword	QueryValue	HKCU\AppEvents\Schemes\Apps\.Default\MenuCommand\.Current\(Default)	SUCCESS	""
155	1...	Winword	OpenKey	HKLM\Software\Microsoft\Office\9.0\Word\Security	NOTFOU...	
156	1...	Winword	OpenKey	0xC29EE690\Microsoft\Office\9.0\Word\Security	NOTFOU...	
157	1...	Winword	QueryVal...	0xC29E72F0\DontTrustInstalledFiles	NOTFOU...	
158	1...	Winword	OpenKey	HKLM\Software\Microsoft\VBA\Trusted	NOTFOU...	
159	1...	Winword	OpenKey	HKCU\Software\Microsoft\VBA\Trusted	SUCCESS	hKey: 0xC1936810
160	1...	Winword	EnumValue	HKCU\Software\Microsoft\VBA\Trusted	NOMORE	
161	1...	Winword	CloseKey	HKCU\Software\Microsoft\VBA\Trusted	SUCCESS	
162	1...	Winword	OpenKey	HKLM\Software\Microsoft\Windows\CurrentVersion	SUCCESS	hKey: 0xC192A600
163	1...	Winword	QueryVal...	HKLM\Software\Microsoft\Windows\CurrentVersion\SubVersionNumber	SUCCESS	20 0
164	1...	Winword	CloseKey	HKLM\Software\Microsoft\Windows\CurrentVersion	SUCCESS	
165	1...	Winword	OpenKey	HKLM\Software\Microsoft\VBA\Trusted	NOTFOU...	
166	1...	Winword	OpenKey	HKCU\Software\Microsoft\VBA\Trusted	SUCCESS	hKey: 0xC29ED140
167	1...	Winword	EnumValue	HKCU\Software\Microsoft\VBA\Trusted	NOMORE	
168	1...	Winword	CloseKey	HKCU\Software\Microsoft\VBA\Trusted	SUCCESS	
169	1...	Winword	OpenKey	0xC29EE690\Microsoft\Office\9.0\Word\Security	NOTFOU...	
170	1...	Winword	QueryVal...	0xC29E72F0\Level	SUCCESS	0x2
171	1...	Winword	SetValueEx	0xC29E72F0\Level	SUCCESS	0x3
172	1...	Winword	OpenKey	0xC29EE690\Microsoft\Office\9.0\Word\Security	NOTFOU...	
173	1...	Winword	QueryVal...	0xC29E72F0\DontTrustInstalledFiles	NOTFOU...	

Figure 6. Monitoring Registry During Changes to Word Security Settings.

Trend Micro has an excellent article, "Safe Computing Guide", that outlines disabling scripting capabilities on Windows 9X machines. Certainly disabling WSH is a wise option in many environments. There may be a temptation to simply change file associations so that any .vbs, .js, or .wsh file will open in Notepad. However, the file associations are under the control of registry settings and are easily discovered.

Using Registry Monitor we can quickly find the registry settings associated with how .wsh file open. As shown in Figure 7 and 8, one simply brings Registry Monitor into action while making a change in the type application used to perform an action.

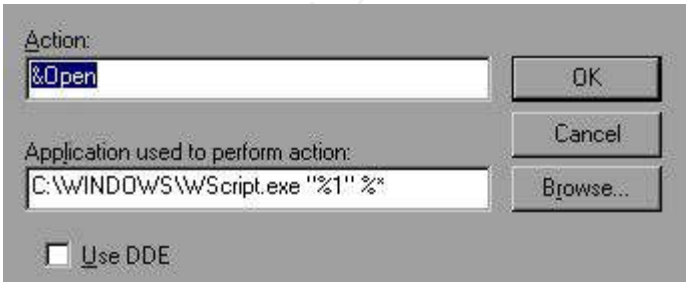


Figure 7. Opened Dialog Box for Editing Action for Wscript.

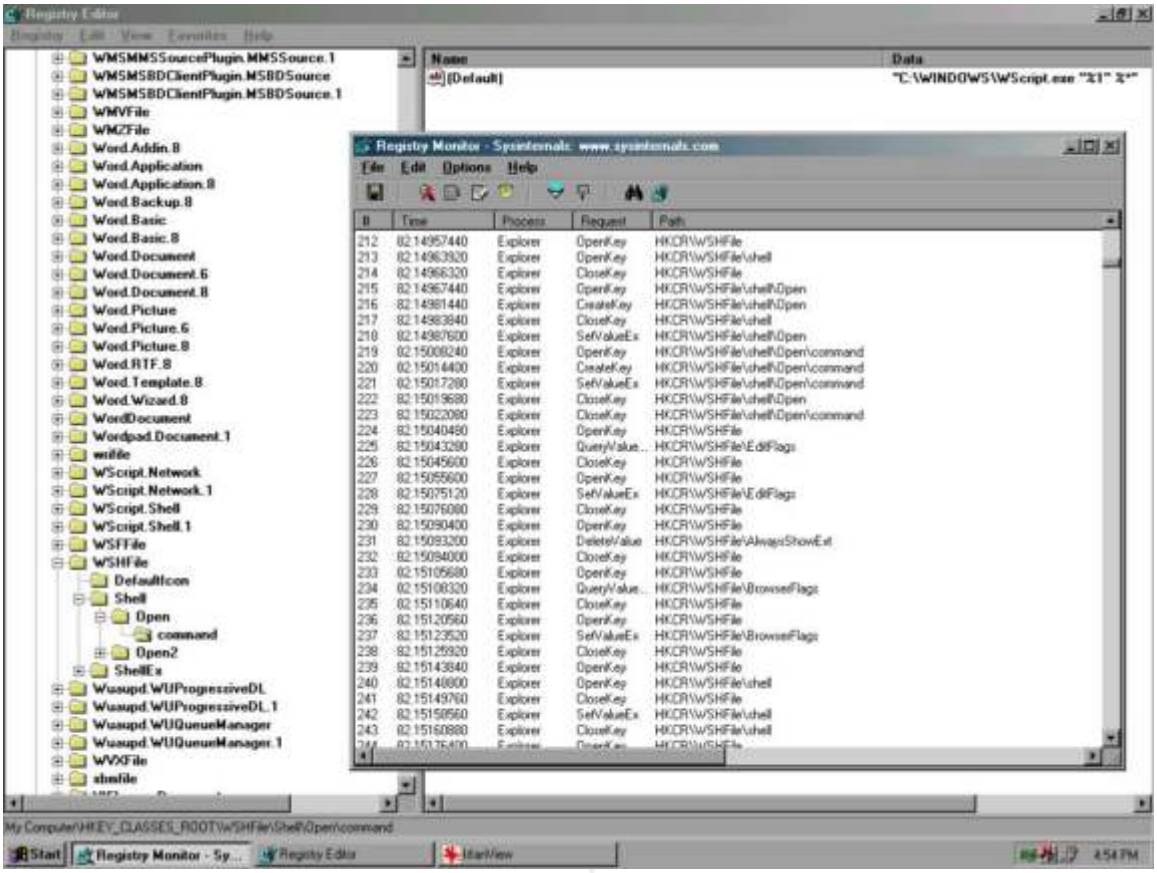


Figure 8. Finding Registry Settings Using Registry Monitor.

Changing the .vbs, .js, and .whs associations would be effective against some malicious code. However, anything that a script can do, an executable or ActiveX control can accomplish. So, the registry and the associations are still open to being attacked and to being changed. While the coding experience is not as brief, an executable can be written to make the same changes the concept script wrote.

In many environments, systems administrators can no longer lock all the doorways and limit access to entry of malicious code or to destructive changes caused by user activities on a Windows 9X machine. Locking the diskette, CDROM drives, and removing administrative tools may not help in the face of the Internet, the World's largest repository of code. If Internet activities are limited, enhanced email remains a doorway. We can attempt to limit the abilities of Macro and Script enhanced application at the expense of reducing or eliminating their utility. However, the real issues with Windows 9X are:

1. On their local machines, users are administrators and all code runs with the power of an administrator or the system.
2. In many environments, we can no longer deny users access to unapproved code.
3. The registry is not secure.
4. The registry contains many settings vital to the security of the machine.
5. Code may be utilized to lower or eliminate registry settings related to security.

Serious consideration should be given to upgrading or replacing Windows 9X machines that need enhanced email, Internet connectivity, enhanced Office applications, or powerful scripting capabilities. The first machines considered for upgrades or replacement might be laptop and notebook computers that operate both inside and outside network firewalls. Moving to properly configured Windows 2000 machines would help eliminate many malicious code problems.[see Note 2]

The concept script utilized in this paper was limited to changing registry settings related to the Internet Explorer Zones. These changes weaken a system to possible exposures to other code. This paper points to the need for further research into staged or layered malicious code. With staged or layered malicious code, one layer or stage can weaken, suppress, or disable large portions of a machine's warning and defense systems. Other layers or stages of code that run with relative impunity follow the initial layer or stage.

Notes 1. The words "malicious" and "malicious code" are used in this paper to convey "bad intent" or "harmful". There has been much discussion and debate on the proper term or terms to use in regard to such code. That debate is outside the scope of this paper.

Notes 2. The author can not, at this time, recommended Windows XP due to its enhancements to support Microsoft's Passport and .Net services.

[BA] Briney, Andy. "2001 Industry Survey". Information Security (Volume 4, Number 10) October 2001.

[CM] Carlisle, Martin C. and Struder, Scott D. "Reinforcing Dialog-Based Security." (June 2001) URL: http://www.securityfocus.com/datta/library/ieeesmc2001.pdf (15 Sept 2001).

[CA] Clinick, Andrew. Microsoft Corporation. "Providing a Secure eXPerience" (8 Oct 2001) URL: http://msdn.microsoft.com/scripting. (17 October 2001).

[HK1] Hayashi, Kaoru. Symantec Security Response. "VBS.Merlin.C@mm." (17 July 2001) URL: http://www.symantec.com/avcenter/venc/data/vbs.merlin.b@mm.html (1 Nov 2001).

[HK2] Hayashi, Kaoru. Symantec Security Response. "VBS.Merlin.C@mm." (8 Aug 2001) URL: http://www.sarc.com/avcenter/venc/pf/vbs.merlin.c@mm.html (1 Nov 2001).

[LE] Lippert, Eric. "Windows Script Host: New Code-Signing Features Protect Against Malicious Scripts." MSDN Magazine. (2001) URL: http://msdn.microsoft.com/library/en-us/dnmag01/html/wsh.asp. (10 Nov 2001).

[MC] Microsoft Corporation. "Widows Scripting Host: A Universal Host for Scripting Languages." (also referenced as, "Microsoft Windows Scripting Host White Paper"). URL: http://www.msdn.microsoft.com/scripting (21 Sept 2001).

[MP1] Microsoft Press. Microsoft Internet Explorer Resource Kit. Redmond, Washington: Microsoft Corporation, 1998.

[MP2] Microsoft Press. Microsoft Windows 95 Resource Kit. Redmond, Washington: Microsoft Press, 1995.

[MPSS1] Microsoft Product Support Services. "Information About the Unsafe File List in Internet Explorer 6 (Q291369)." URL: http://www.microsoft.com/technet (10 Oct 2001).

[MPSS2] Microsoft Product Support Services. "XCLN: How to Modify the Behavior of the Attachment Security Warning in Outlook 2000 SR-1 (Q259514)." (22 June 2001) URL: http://support.microsoft.com/support/kb/Q259/5/14.asp. (10 Oct 2001)

[MT] Microsoft TechNet. "Chapter 20: Internet Access & Tools." Microsoft Windows 98 Resource Kit. Redmond, Washington: Microsoft Corporation, Technical Information CD, April 2001.

[MTI] Microsoft Technical Information. "Description of Internet Explorer Zones Registry Entries (Q182569)." URL: http://www.microsoft.com/technet (10 Oct. 2001).

[PC1] Personal Correspondence. Scott. Microsoft Security Response Center. (16 Sept. 2001).

[PC2] Personal Correspondence. Russ Cooper. NTBUGTRAQ. (17 Sept 2001 and 18 Sept 2001).

[PR] Petrusha, Ron. Inside the Widows 95 Registry. Sebastopol, CA: O'Reilly & Associates. Inc, 1996.

[RM] Russinovich, Mark and Cogswell, Bryce. Registry Monitor for Windows NT / 9x, Version 4.32. (2001) URL: http:\www.sysinternals.com (15 Sept 2001)

[SS] Schnoll, Scott. "The History of Microsoft Internet Explorer." 2001. URL: http://www.nwnetworks.com/iesc.html (2 Oct. 2001).

[ST] SecuriTeam.com. "Norton Antivirus 2002 Security Flaws." (17 June 2001) URL: http://www.securiteam.com/windowsntfocus/5GP0C2A4UO.html (12 Nov 2001)

[SW] Stanek, William R. Windows 2000 Scripting Bible. Foster City, CA: IDG Books Worldwide, Inc (2000)

[SV] Svajcer, Vanja. "VBA Implementation & Virus Risks." Virus Bulletin Conference (Sept 2000) URL: www.virusbtn.com/vb2000/programme/index.html (3 Oct 2001)

[SG] Szappanos, Gabor. "Micro Virus Protection in Microsoft Office Line, Part Two." (26 Sept 2001) URL: http://www.securityfocus.com/infocus/1484. (10 Oct 2001)

[TM] Trend Micro. "Safe Computing Guide." (2001) URL: http://www.antivirus.com/pc-cillin/vinfo/safe_computing/ (18 Sept 2001)