Global Information Assurance Certification Paper

Configuring Internet Explorer Security Zones:
A New Tool for the Security Community

by
Ken Barber
Submitted to the SANS Institute for the GIAC Security Essentials Certification version 1.3

Abstract

Microsoft's Internet Explorer (hereafter, **IE**) web browser is a major source of headaches for security administrators. Even when leaving aside the topic of endless patches that must be applied to the vulnerability *du jour* – which is not the subject of this paper – we are still faced with enormous security risks from **active content** technologies such as Java, Javascript and ActiveX. Recognizing these risks, Microsoft introduced a device called **security zones** into version 4.0 of its browser product that continues to be an integral part of IE to this day. Basically, this device divides a user's Web "world" into zones which are presumed to have various levels of safety: some sites (such as those contained on a corporate intranet, for instance) are far less likely to contain malicious code than others.

This paper will review the work of others in discussing the risks inherent in each of the active content technologies, and the very different ways in which they approach security. Then it will gather into one place all of the information that the author could find regarding the meanings and implications of all but one of IE's security zone settings. After that, we shall discuss Microsoft's **System Policy Editor** tool for Windows NT and how it could have been used to quickly and easily enforce users' IE security zone settings throughout an enterprise, had Microsoft only provided a **policy editor template** for the IE security zones. Finally (and admittedly somewhat belatedly) a template to do just that, written by the paper's author, shall be presented to the security community.

Table of Contents

Terms of art used in this paper

Hacker

This is a word that is in danger of becoming redefined through constant misuse by the news media. To quote from the Webopedia's definition of this word:

> "A slang term for a computer enthusiast, i.e., a person who enjoys learning programming languages and computer systems and can often be considered an expert on the subject(s). ... The pejorative sense of *hacker* is becoming more prominent **largely because the popular press has coopted the term** [emphasis mine] to refer to individuals who gain unauthorized access to computer systems...." (Webopedia)

In other words, the term *hacker* is increasingly being misused to mean *cracker*, which is the correct term for a hacker with malicious intent. Many careful writers use the term *malicious hacker* but this term takes up a lot of space and is cumbersome when speaking (this author recently heard one speaker deal with this by using the term *evil hacker*, which is much easier to say). Since this author has become accustomed to using the term *'hacker'* (i.e., enclosing the misused word in single quotes) in casual emails, this paper shall continue that convention.

Where any form of the word *hack* (as in "hacking the registry") appears in this paper unenclosed, its meaning is to be interpreted in the benign sense of the word.

Genderless third-person pronouns

There is a growing movement among the Internet community to use "Gender-Neutral Pronouns" to replace clumsy constructions such as "his/her" when referring to generic persons. Several standards have been proposed; the one to which this author has become accustomed is the word *zir* to replace *him or her*, *his or her*, etc. While that particular proposal also uses *zie* to replace *he or she* this author finds *s/he* to be easier on the reader, and shall use it instead.

More information on gender-neutral pronouns is available at http://www.aetherlumina.com/gnp.

An Introduction to Active Content Technologies

Active Content is the term that Microsoft uses to describe the act of attaching computer programs to Web pages, which are then executed on the client machine. Two competing technologies exist for this purpose: **Java**, which was developed by Sun Microsystems, and **ActiveX**, which was developed by Microsoft Corporation. (Felton) A third technology known as **JavaScript** also exists, developed by Netscape Communications. Though their names are similar, Java and JavaScript are very different critters.

Both ActiveX and Java function through compiled programs known as **controls** (in the case of ActiveX) or **applets** (in the case of Java) that exist as separate files but are downloaded at the same time as the Web pages that contain them, and are executed by the Web browser through commands embedded in the page's HTML code.

Unlike Java and ActiveX, which are compiled executables, JavaScript is basically an extension to the Hypertext Markup Language that was developed by Netscape Communications in an attempt to bring some of Java's functionality to Web designers who were not programmers:

> The hope was that this new language ... would be simpler to understand than Java and would provide web designers with the ability to include interactive elements in their pages without having to master a complex programming language....
>
> JavaScript is easily integrated into the HTML code for a web page. ... Unlike Java or CGI, JavaScript does not require a separate file for code.... (Dybka)

Security Implications of Active Content

The inherent hazards of active content should be obvious. Users are automatically downloading and running programs that were written by strangers, often without even being aware that they are doing so. Automatically downloading and running programs from unknown sources on one's computer is never a good idea. Unfortunately, the Internet has evolved to the point where most people expect to be able to do things on the Web (such as fill out a form) that simply cannot be done without the presence of active content.

Both Microsoft and Sun have developed measures to minimize the dangers inherent in running "anonymous code" on one's computer. As we shall soon see, the two companies have taken very different approaches, with varying levels of adequacy, to the security problem of active content.

JavaScript is somewhat safer than the compiled languages (though by no means should it be considered completely safe). For one thing, since it is an interpreted language its

source code is readily readable by anyone who chooses to do so, making it nearly impossible to hide malicious code. It is also severely limited in what it can do to a filesystem.

Java security vs. ActiveX security

ActiveX controls are digitally signed by their authors, and optionally by a certificate authority as well. While the technology behind digital signatures is sound, it only confirms an author's identity. It does not confirm that the author is an honest person, or even that s/he knows how to write secure code. Further, Microsoft's implementation of digital signatures breaks the technology in significant ways: for instance, ActiveX does not check for revoked certificates, meaning that an attacker who obtains a certificate through fraudulent means (which has happened) will continue to have his malicious control accepted by browsers even after he has been found out. It also means that a malicious website operator could exploit a security-related bug in an older version of a legitimate control from a reputable author. (Stein, Hopwood)

The biggest problem with ActiveX security, however, is that it ultimately places the responsibility for security decisions on security's weakest link: the end user. In its default configuration, the only protection that IE gives against suspicious controls (i.e., invalid certificates or unsigned controls) is a warning dialog – and we all know that most users will simply click "yes" when presented with such a choice, often without even reading the warning! More resourceful users can even find helpful Microsoft's Knowledge Base articles (such as Q164004) which will tell them how to reduce or disable their security settings so they won't be bothered anymore by those annoying warning messages.

While Java applets can also be digitally signed (with the same security implications as discussed above) unsigned applets do not force users into "all or nothing" decisions but are instead run in a **sandbox** where their actions are restricted to those which are deemed safe.

The Secure Internet Programming team at Princeton University summed it up nicely:

> ActiveX security relies entirely on human judgment. ... You have two choices: either accept the program and let it do whatever it wants on your machine, or reject it completely. ActiveX security relies on you to make correct decisions about which programs to accept....
>
> Java security relies entirely on software technology. Java accepts all downloaded programs and runs them within a security "sandbox." ... Java security relies on the software implementing the sandbox to work correctly. (Felton)

We should note that JavaScript security also relies on software technology. All three active content technologies are vulnerable to bugs in the software that implements them,

and dozens of security-related bugs have been found in each.

However, if it were possible to write absolutely bullet-proof engines for each of the three technologies, ActiveX would still pose significant threats to security because of the virtually unlimited power it has on a system, and the ease with which users can be persuaded to grant that power. This was made clear in 1997 when a hacker group called the Chaos Computer Club demonstrated an ActiveX control on live TV that causes Quicken to silently transfer money from the victim's bank account! (Wingfield)

Microsoft's response to that incident was typical of its blame-the-user mentality:

> "What this incident tells us is you cannot take candy from strangers. The thing I'm hoping users get out of this is that they should not be running any executable code that is anonymous." (ibid.)

I don't know how the average user can reasonably be expected to make such distinctions, let alone know what "anonymous code" even means.

The fact that any program of a malicious Web site operator's choosing can be made to run on virtually any system using Internet Explorer should give every security-minded manager cause to consider the implications of running Windows workstations – from which IE is difficult to impossible to remove – in a corporate network. The only reason there have not been any major 'hacking' events (as of this writing) that exploit ActiveX is simply because those with the skills to do so have not chosen to apply their skills in this area. (Felten)

The Internet Explorer security zones

In version 4.0 of Internet Explorer Microsoft introduced a tool known as **security zones** in an attempt to reduce the exposure to risk that active content presents. The assumption behind the security zones model is that some web sites (such as a corporate intranet) are safer than others. Accordingly, the user's world is classified into four zones: **Internet, Local Intranet, Trusted Sites,** and **Restricted Sites.** Within each of these zones are 22 user-configurable settings governing what kinds of actions the user will permit incoming Web pages to perform on zir machine. A fifth zone, **Local Machine,** is not considered user-accessible since it can be configured only by hacking the registry. (Schnoll)

The first zone is where all Web sites exist by default, except for a few which meet a somewhat convoluted set of rules for inclusion in the Local Intranet zone. Users are able to add individual sites to any zone other than the Internet zone, depending on the level of trust they place in each site.

The Trusted Sites zone is one in which users can place the names of sites which are not on the local intranet, but are still known to be trustworthy. This zone has low security settings by default, and it is reportedly possible to configure this zone to require servers to

authenticate themselves through SSL before trusting them. (Schnoll)

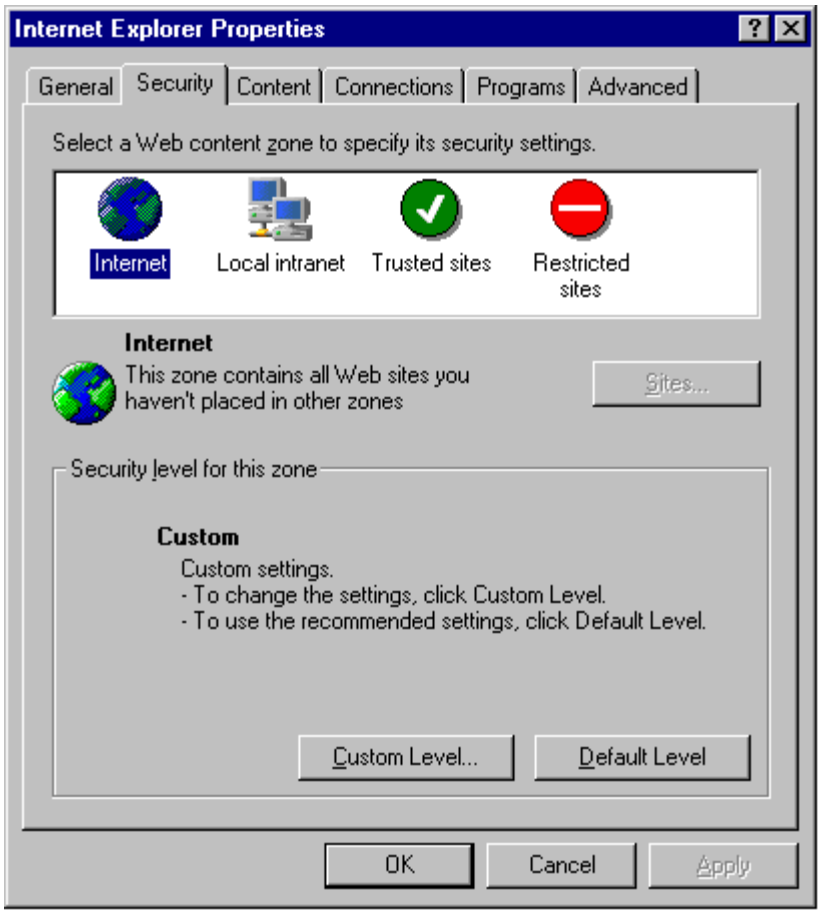
The Restricted Sites zone is a somewhat amusing example of Microsoft's naïveté in matters pertaining to network security. According to Scott Schnoll's otherwise excellent paper:

> You can liken this zone to a jail. When used properly, the Restricted Sites zone provide [sic] a mechanism for containing web sites that may cause damage to or steal information from your system or your network. Obviously the best protection would be to simply not visit the sites.... (Schnoll)

It should be obvious to any astute person that every web site, except for a small handful known to the organization, may cause damage to you, or steal information from you! Therefore, every unknown web site should be in the restricted zone – but sites are not members this zone until they've been manually added. The futility (not to mention stupidity) of doing this is roughly equivalent to a gag in a Three Stooges movie.

The Local Intranet zone and the rules by which a site gets automatically placed there are beyond the scope of this paper. The inquisitive reader is hereby directed to Schnoll's paper or the Microsoft Technet article in the List of References.

The Internet Explorer Security Zones settings



As mentioned earlier, there are 22 user-configurable settings (24 in IE 6) for each of the security zones in Internet Explorer. This paper covers 21 of them in detail. These are accessed through IE's Tools menu and thence from the Internet Options submenu.

This opens a window titled Internet Explorer Properties (see picture) with a number of tabbed pages. Choosing the Security tab will display the four security zones, with different options depending on the particular zone chosen. Each of the zones has a **Custom Level** button that will open a new window with the 22 settings in it.

Most of these settings have three choices: Enable the action, disable the action, or prompt the user with a yes/no decision. For instance, persistent cookies can be enabled, disabled, or ask the user whether to accept a cookie every time one is offered.

The third choice (Prompt) is really only useful for testing or educational purposes. In the real world, dialogs constantly popping up asking users what to do are not only annoying, but almost always result in the user making the more dangerous of the two choices.

An excellent guide to these settings can be found in the TechNet paper (see "TechNet" in the List of References). Administrators who need to configure the IE security zones are strongly encouraged to get, read and study that document: it contains much useful information that would be senseless to repeat *ad infinitum* here.

The 22 Security Zone settings are grouped under the following headings, which we shall consider in turn:

- ActiveX Controls and Plug-ins
- Cookies
- Downloads
- Java
- Miscellaneous
- Scripting
- User Authentication

ActiveX Controls and Plug-ins

The following settings appear in this group:

- Download signed ActiveX controls
- Download unsigned ActiveX controls
- Initialize and script ActiveX controls not marked as safe
- Run ActiveX controls and plug-ins
- Script ActiveX controls marked safe for scripting

ActiveX controls are small programs that can be run from a Web browser. Typically they appear as some kind of an interactive graphic on which the user clicks to run the program; however, they can also be called from a script and run automatically without any action on the part of the user (or even without the user's knowledge).

ActiveX controls can exist on the user's filesystem, or they can be downloaded along with the Web page that contains them. There are no limitations to what an ActiveX control can do once it has been permitted to run: they can do anything the user has permission to do, including erasing files and modifying the operating system: indeed, this is how Microsoft's Windows Update service functions.

Clearly, these critters have an enormous potential for mischief. If a particular control has the ability to write data of the user's choosing to the local filesystem (and many do have that ability), then if that control were launched with a script it could be subverted to change operating system files without the user's knowledge. Such controls are considered "not safe for scripting" and are supposed to be marked accordingly.

Whether or not a control is marked "safe for scripting" is entirely up to the control's author. A 'hacker' can write controls that place Trojans, back doors and password sniffers on a victim's computer, and falsely mark them "safe for scripting." It is also possible for an honest but inexperienced programmer to mistakenly mark a control "safe for scripting" when it is not safe, which a 'hacker' can then exploit for malicious purposes. This has actually happened. (Hopwood)

Most 'hackers' won't go to the trouble to sign a control and then set up a phony certificate authority to verify the signature, but it is certainly possible to do so, and has been done as a proof of concept. (McLain) It is also well within the ability of an oppressive government to write controls that secretly plant back doors and password sniffers on one's machine, and have them signed and authenticated. Indeed, the United States of America's National Security Agency is reportedly already doing this (Byrne).

Some of these settings can override the others. For instance, if **Initialize and Script ActiveX controls that are not marked as safe** is enabled, all of the other settings are also enabled (TechNet).

Recommendations:

- Do not enable **Download unsigned ActiveX controls** in the Internet zone. It is probably safe to enable this in the Local Intranet zone if you trust all of your in-house developers, and you are certain you have configured the criteria for sites' inclusion in the Local Intranet zone correctly (not covered in this paper; see Schnoll and TechNet in the List of References).
- Do not enable **Download signed ActiveX controls** in the Internet zone unless your organization has absolutely nothing anywhere on its network worth stealing! Neither should you enable this control if you have any reason to fear your government.
- Do not enable **Initialize and script ActiveX controls not marked as safe** in any zone. In the Microsoft's words:

  > An example of a control that poses a security risk at initialization time would be a data compression/decompression control. If the user pointed this control to a remote, compressed file that contained a Trojan-horse copy of a system file (such as Kernel.dll) and requested that the control decompress this file, system security could be breached. (MSDN-ActiveX)

  When the Borg Collective itself warns us that something isn't safe and could result in a hull breach, we should probably sit up and take notice.
- The **Run ActiveX controls and plug-ins** setting has an option to allow only administrator-approved controls to run. See the topic "Managing ActiveX Controls" in the help system of the IEAK to find out how to create and maintain this list.

Cookies

- Allow cookies that are stored on your computer
- Allow per-session cookies

This paper assumes that the reader is already familiar with cookies, what they do, how they work and the risks that they pose to user privacy. The reader who is new to this subject is encouraged to enter the words "browser cookies" into a search engine – and settle down to a nice long winter of reading because there are a lot of people who have

something to say about this subject.

Per-session cookies are cookies that are stored in memory but not on the hard drive; they live as long as the browser is running.

The bottom line here is that so many Web sites now use cookies that no network administrator can get away with disabling them without having a user revolt. It won't even work to allow per-session cookies while disabling stored cookies; many Web sites will sense this and will still refuse to function until persistent cookies are enabled.

This author's favorite method of dealing with this is to use a Unix (or one of its derivatives) machine to store the Cookies directory – and make that directory a symlink to /dev/null. The browser cheerfully accepts cookies all day long, and web sites that demand persistent cookies are happy – but the cookies promptly disappear into a black hole as soon as the browser is closed!

Downloads

- Download files
- Download fonts

Little needs to be said here. Whether to allow downloading files is a policy decision that each organization needs to make on its own. Some have claimed that font files can contain executables; however, this author was unable to find authoritative corroboration of that claim. Let the administrator beware.

Java

The Java group has only one item –

- Java permissions

– which has five possible settings: Low, Medium, High, Disable Java and Custom. Unfortunately, there is not room in this paper to discuss the rich set of permissions that can be configured under "Custom." Further, this author has been unable to find a way to set any of these custom settings using the tool described below. Herein lies grist for a future paper, should anyone choose to write it.

Miscellaneous

- Access data sources across domains
- Drag and drop or copy and paste files
- Installation of desktop items
- Launching programs and files in an IFRAME
- Navigate sub-frames across domains
- Software channel permissions
- Submit nonencrypted form data
- Userdata persistence

According to Microsoft: (TechNet)

Access data sources across domains refers to accessing databases that reside on a different server from the one hosting the Web page.

Drag and drop or copy and paste files refers to whether users will be allowed to perform these actions on files that exist inside of a Web page. How this differs (or whether it differs) from plain old downloading isn't explained.

It isn't clear to this author just what a **desktop item** is. Apparently it has something to do with the Active Desktop that was part of IE 4.

An **IFRAME** is a floating Windows Explorer (i.e., Microsoft's file manager program) frame that can be made to appear in a Web page with the following bit of HTML code:

`<IFRAME src="\\server\share\directory"></IFRAME>` (Q232077)

Files that are within the referenced directory can be launched from a browser just as if the user were launching them from Windows Explorer. This can obviously be useful in a corporate intranet, but it would be foolish to enable it in the Internet zone.

Software Channel is apparently a throwback to the bad ol' days of IE 4.0 when Microsoft was first experimenting with the idea of automatically downloading and installing software updates through a user's Web browser. The TechNet article (see "TechNet" in the List of References) speaks of settings that notify users of updates via email and automatically download and install them, or automatically download but not install them, and so forth.

Submit non-encrypted form data refers to whether users will be able to fill out on-line forms on non-SSL Web pages. The threat here is to the user's privacy rather than the security of the user's computer or the internal network to which it is attached.

Userdata Persistence is a feature that was introduced in IE 5.0 and can be thought of as "super-cookies" (or perhaps "cookies on steroids," "monster cookies" or "teenage mutant kung-fu cookies"). Like cookies, it stores user data across sessions. The difference is that each of these "small files of personal information" (Microsoft's words) can be as large as 128K in size! (MSDN-Persistence) The issues here are the same as the issues for cookies, and will be different for each organization.

Scripting

- Active scripting
- Allow paste operations via script
- Scripting of Java applets

Active scripting is Microsoft-speak for "scripts that can be written in either of two languages: standards-based JavaScript or (preferably) our proprietary VBscript which will enable us to lock you into using only our products." (Berlind) You cannot enable or disable one without also enabling or disabling the other. The other two settings mean exactly what they say: allowing scripts to cut, copy or paste material to or from the Clipboard, and allowing scripts to launch Java applets.

There are many, many Web pages out there that rely on JavaScript, and few organizations today can survive the political fallout from disabling it (picture villagers storming the IT department's doors with pitchforks and torches). While there is some security risk in JavaScript, it is minimal compared to other technologies discussed in this paper. However, enabling JavaScript also enables VBScript, and the security implications of the latter are unknown to this author.

It is not wise to enable **Scripting of Java applets** or **Allow paste operations via script** from the Internet zone.

User Authentication

- Logon

This setting has four mutually exclusive options:

- Anonymous logon
- Automatic logon only in Intranet zone
- Automatic logon with current user name and password
- Prompt for user name and password

In order to fully understand the security implications of these choices it is necessary to understand the weaknesses in Microsoft's implementation of the CIFS (Common Internet File System, aka SMB) protocol and the NTLM authentication method. Each has serious security vulnerabilities, and Microsoft uses them together to transfer files over a network. (Hobbit)

Both of the **Automatic logon** choices send the user's NTLM username and password over the wire – automatically, and transparently to the user – any time a Web server requests it. Since NTLM passwords are ridiculously easy to crack, the potential for mischief by operators of rogue Web sites is obvious. If NTLM authentication is necessary for an organization's internal Web site to function, then it should only be enabled in the Intranet zone. Using that option (i.e., **Automatic logon only in Intranet zone**) causes a different method known as HTTP Authentication to be used in the other three zones. HTTP Authentication is the method that **Prompt for user name and password** uses, and choosing it causes it to be used in all zones.

It should also be noted that when HTTP Authentication is used, it caches the username and password for the remainder of the session. (TechNet) This might have security implications in an environment where workers often must leave their workstations while

still logged in.

Anonymous logon disables both authentication methods, using instead the Guest account that is built in to all Microsoft servers to access files via the CIFS protocol. (TechNet) Obviously, this limits the files that can be accessed to those to which Guest has permissions.

Using NT Policy Editor (POLEDIT.EXE)

Having explored in detail 21 of Security Zone settings and learning as much as we can about them, we now turn to the question of how administrators of an NT 4.0 environment can enforce the desired settings enterprise-wide.

Microsoft's solution to this is a ridiculously complex scheme that requires using the IEAK to build a customized version of the browser, which then must be deployed on all desktops in the organization. Within this custom build, the administrator specifies a file in an SMB share on a server somewhere that contains the IE settings the administrator wants to enforce. Then the IEAK Profile Manager is used to build that file, which has a .INS extension. Then, every time a user launches zir browser, it loads its settings from the .INS file. (IEAK-help)

There is a better way. There is a tool known as System Policy Editor (POLEDIT.EXE) that ships with the server versions of Microsoft's operating systems. It enables administrators to create a file that contains certain registry configurations which will be automatically downloaded and applied to the HKEY_LOCAL_MACHINE and HKEY_CURRENT_USER hives every time a user logs on to the domain. This file resides in the NETLOGON shares on domain controllers (the same directories in which the login script is stored) and is named NTCONFIG.POL.

The Internet Explorer Security Zones settings reside in both the HKLM and HKCU hives. Another key in HKLM controls which of these hives Internet Explorer actually uses.

The particular registry keys that System Policy Editor can configure, and the possible values that it can assign to them, are controlled by **Administrative Templates** (.ADM files) that can be loaded into the Policy Editor before creating or modifying NTCONFIG.POL. Several of these .ADM files have been written over the years, and they have come to be very valuable to administrators seeking to enforce one policy or another. Any number of .ADM files can be loaded into a single session of the Policy Editor, making it a truly flexible tool.

If an administrator, for instance, wishes to set a certain screensaver to activate after a certain period of inactivity on all workstations, or control whether users can browse Network Neighborhood or open the Control Panel, there are .ADM files for that that ship with the Policy Editor. If an administrator also wishes to set the default file locations for Microsoft Office products, there is a .ADM file for that that ships with the Office Resource Kit, and all of these can be loaded at the same time and packaged into a single

NTCONFIG.POL file.

So, if an administrator wishes to enforce policy regarding the IE Security zones, all s/he must do is find a .ADM file for that and load it, too into System Policy Editor, right? Alas, this author has been unable to find any such critter.

Introducing a new tool to the security community: IESEC.ADM

Detailed instructions on using the NT System Policy Editor and creating custom administrative templates (.ADM files) for it are found in the Microsoft white paper (see "Microsoft white paper" in the list of references). Details of the registry keys and values for the IE Security Zone settings can be found in Q182569.

This author has written an incomplete .ADM file that can be used in System Policy Editor to configure the Security Zones settings. It appears in the Appendix.

Caveats

1. This template configures the Internet zone only.
2. This template modifies the user portion of the registry (HKEY_CURRENT_USER) only; no modifications to the machine's hive are made.
3. Nothing in this template prevents users from changing their security zones settings after they have logged on; it only sets them at user login.
4. To prevent users from changing their settings, this template would have to be extended or modified to affect the HKLM hive. Another key, which is documented in Q182569, would also have to be included in the template. That key is:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows
\CurrentVersion\Internet Settings Security_HKLM_only
```

Others are invited to extend this tool as needed to fit their needs.

Conclusion

It is this author's opinion that Internet Explorer cannot be made safe, and probably will continue to be that way for the foreseeable future. ActiveX is an inherently dangerous technology on which Bill Gates has "bet the farm" and as such it isn't likely to disappear until Redmond freezes over and penguins start roosting there.

This paper grew from an effort by this author to secure the IE implementation at his workplace as much as is possible – and found Microsoft's cryptic explanations of the various Security Zones settings to be virtually useless. It is hoped that the information discovered over the last eight months and presented here (which the author believes exists in no other single paper) will be useful to security-minded administrators everywhere.

For those brave souls who dare to challenge the Microsoft orthodoxy, it is supposedly possible to remove Internet Explorer from Windows entirely, leaving open the option of using Mozilla, Netscape or Opera instead as an organization-wide standard. The curious reader will find this information at http://www.98lite.net.

Good luck getting such a proposal past upper management! In the meantime, you can use the information in this paper to make your network more secure than it otherwise would have been.

List of References

Berlind, David. "Standards Can Put You in Control." ZDNet Tech Update: Reality Check 10 Jan 2002 URL: http://techupdate.zdnet.com/techupdate/stories/main/0,14179,2837626,00.html (05 Feb 2002)

Byrne, Peter. "The Spybots Among Us." San Francisco Weekly December 19-25, 2001 (2001): 18-27 URL: http://www.sfweekly.com/issues/2001-12-19/feature.html/1/index.html

Dybka, Jill and Shoemaker, Leigh. "Java vs. JavaScript: A Detailed Comparison." URL: http://itc.utk.edu/itc/clearinghouse/java/jvsjs.html (28 Jan 2002)

Felton, Edward. "Security Tradeoffs: Java vs. ActiveX." 28 April 1997 URL: http://www.cs.princeton.edu/sip/faq/java-vs-activex.html (29 Jan 2002)

Hobbit. "Common Insecurities Fail Scrutiny." Jan 1997 This definitive work is available at fine archives all over the Internet. Some of them are:
http://www.signaltonoise.net/library/cifs.htm (9 Feb 2002)
http://downloads.securityfocus.com/library/cifs.txt (9 Feb 2002)

Hopwood, David. "A comparison between Java and ActiveX Security." 10 Oct 1997 URL: http://www.users.zetnet.co.uk/hopwood/papers/compsec97.html (30 Dec 2001)

IEAK-Help. "How to Use the IEAK Profile Manager." Microsoft Corporation, Internet Explorer Administration Kit 5.5 Help System.

McLain, Fred. "The Exploder Control Frequently Asked Questions (FAQ)." 7 Feb 1997 URL: http://www.halcyon.com/mclain/ActiveX/Exploder/FAQ.htm (31 Dec 2001)

Microsoft white paper. "Implementing Policies and Profiles for Windows NT 4.0." URL: http://www.microsoft.com/ntserver/techresources/management/prof_policies.asp (12 Feb 2002)

MSDN-ActiveX. "Safe Initialization and Scripting for ActiveX Controls." URL: http://www.msdn.microsoft.com/workshop/components/activex/safety.asp (5 Feb 2002)

MSDN-Persistence. "Introduction to Persistence." URL: http://msdn.microsoft.com/workshop/author/persistence/overview.asp (7 Feb 2002)

Q[six-digit number]. Microsoft knowledge-base article. URLs change; to access these, go to http://support.microsoft.com and enter the "Q-number" into that page's search window. Knowledge-base articles referenced in this paper are:

Q182569: "Description of Internet Explorer Security Zones Registry Entries."
Q164004: "Microsoft Internet Explorer on High Safety Stops Subform Display"
Q232077: "Executing Files by Hyperlink and the File Download Dialog Box."

Schnoll, Scott. "Internet Explorer Security Zones." URL: http://www.nwnetworks.com/iezones.htm (6 Jul 2001)

Stein, Lincoln and Stewart, John. "The World Wide Web Security FAQ." Version 3.1.1, September 12, 2001 URL: http://www.w3.org/Security/Faq/ (29 Jan 2002)

TechNet, Microsoft. "Security Zones and Permission-Based Security for Microsoft Virtual Machine." URL: http://www.microsoft.com/TechNet/prodtechnol/windows2000serv/reskit/ie50rg/part1/ch07zone.asp (5 Feb 2002)

Webopedia. URL: http://www.webopedia.com (13 Feb 2002)

Wingfield, Nick. "How Safe Is Microsoft's ActiveX?" 11 Feb 1997 URL:

http://www.eto.org.uk/faq/faqactvx.htm (30 Jan 2002)

Appendix

Contents of the `IESEC.ADM` file

```
; Internet Settings Policy Template File
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
CLASS USER ;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

CATEGORY !!Zone3Security
KEYNAME "Software\Microsoft\Windows\CurrentVersion\Internet
Settings\Zones\3"

        POLICY !!ActiveX
                PART !!DownloadSignedControls DROPDOWNLIST
                VALUENAME "1001"
                ITEMLIST
                        NAME !!Disable VALUE NUMERIC 3 DEFAULT
                        NAME !!Enable VALUE NUMERIC 0
                        NAME !!Prompt VALUE NUMERIC 1
                END ITEMLIST
                END PART

                PART !!DownloadUnSignedControls DROPDOWNLIST
                VALUENAME "1004"
                ITEMLIST
                        NAME !!Disable VALUE NUMERIC 3 DEFAULT
                        NAME !!Enable VALUE NUMERIC 0
                        NAME !!Prompt VALUE NUMERIC 1
                END ITEMLIST
                END PART

                PART !!InitAndScriptNotMarkedSafe DROPDOWNLIST
                VALUENAME "1201"
                ITEMLIST
                        NAME !!Disable VALUE NUMERIC 3 DEFAULT
                        NAME !!Enable VALUE NUMERIC 0
                        NAME !!Prompt VALUE NUMERIC 1
                END ITEMLIST
                END PART

                PART !!RunActiveXControls DROPDOWNLIST
                VALUENAME "1200"
                ITEMLIST
                        NAME !!AdminApproved VALUE NUMERIC 65536
                        NAME !!Disable VALUE NUMERIC 3
                        NAME !!Enable VALUE NUMERIC 0 DEFAULT
                        NAME !!Prompt VALUE NUMERIC 1
                END ITEMLIST
```

```
        END PART

        PART !!ScriptSafe DROPDOWNLIST
        VALUENAME "1405"
        ITEMLIST
                NAME !!Disable VALUE NUMERIC 3
                NAME !!Enable VALUE NUMERIC 0 DEFAULT
                NAME !!Prompt VALUE NUMERIC 1
        END ITEMLIST
        END PART
END POLICY

POLICY !!Cookies
        PART !!AllowPersistentCookies DROPDOWNLIST
        VALUENAME "1A02"
        ITEMLIST
                NAME !!Disable VALUE NUMERIC 3
                NAME !!Enable VALUE NUMERIC 0 DEFAULT
                NAME !!Prompt VALUE NUMERIC 1
        END ITEMLIST
        END PART

        PART !!AllowSessionCookies DROPDOWNLIST
        VALUENAME "1A03"
        ITEMLIST
                NAME !!Disable VALUE NUMERIC 3
                NAME !!Enable VALUE NUMERIC 0 DEFAULT
                NAME !!Prompt VALUE NUMERIC 1
        END ITEMLIST
        END PART
END POLICY

POLICY !!Downloads
        PART !!Filedownload CHECKBOX
        VALUENAME "1803"
        VALUEON NUMERIC 0
        VALUEOFF NUMERIC 1
        END PART

        PART !!FontDownload DROPDOWNLIST
        VALUENAME "1604"
        ITEMLIST
                NAME !!Disable VALUE NUMERIC 3
                NAME !!Enable VALUE NUMERIC 0 DEFAULT
                NAME !!Prompt VALUE NUMERIC 1
        END ITEMLIST
        END PART
END POLICY

POLICY !!MicrosoftVM
        PART !!JavaPermissions DROPDOWNLIST
        VALUENAME "1C00"
        ITEMLIST
                NAME !!DisableJava VALUE NUMERIC 0
```

```
                        NAME !!HighSafety VALUE NUMERIC 256
DEFAULT
                        NAME !!MedSafety VALUE NUMERIC 512
                        NAME !!LowSafety VALUE NUMERIC 768

                END ITEMLIST
                END PART
        END POLICY

        POLICY !!Miscellaneous
                PART !!CrossDomainData DROPDOWNLIST
                VALUENAME "1406"
                ITEMLIST
                        NAME !!Disable VALUE NUMERIC 3 DEFAULT
                        NAME !!Enable VALUE NUMERIC 0
                        NAME !!Prompt VALUE NUMERIC 1
                END ITEMLIST
                END PART

                PART !!DontPrompt CHECKBOX
                VALUENAME "1A04"
                VALUEON NUMERIC 3
                VALUEOFF NUMERIC 0
                END PART

                PART !!DragDropCopyPaste DROPDOWNLIST
                VALUENAME "1802"
                ITEMLIST
                        NAME !!Disable VALUE NUMERIC 3
                        NAME !!Enable VALUE NUMERIC 0 DEFAULT
                        NAME !!Prompt VALUE NUMERIC 1
                END ITEMLIST
                END PART

                PART !!InstallDeskItems DROPDOWNLIST
                VALUENAME "1800"
                ITEMLIST
                        NAME !!Disable VALUE NUMERIC 3 DEFAULT
                        NAME !!Enable VALUE NUMERIC 0
                        NAME !!Prompt VALUE NUMERIC 1
                END ITEMLIST
                END PART

                PART !!LaunchProgsInIFRAME DROPDOWNLIST
                VALUENAME "1804"
                ITEMLIST
                        NAME !!Disable VALUE NUMERIC 3 DEFAULT
                        NAME !!Enable VALUE NUMERIC 0
                        NAME !!Prompt VALUE NUMERIC 1
                END ITEMLIST
                END PART

                PART !!SubFramesXDomains DROPDOWNLIST
                VALUENAME "1607"
```

```
                ITEMLIST
                        NAME !!Disable VALUE NUMERIC 3 DEFAULT
                        NAME !!Enable VALUE NUMERIC 0
                        NAME !!Prompt VALUE NUMERIC 1
                END ITEMLIST
                END PART

                PART !!SoftwareChannelPerms DROPDOWNLIST
                VALUENAME "1E05"
                ITEMLIST
                        NAME !!HighSafety VALUE NUMERIC 65536
DEFAULT
                        NAME !!LowSafety VALUE NUMERIC 196608
                        NAME !!MedSafety VALUE NUMERIC 131072
                END ITEMLIST
                END PART

                PART !!SubmitNonencryptedForms DROPDOWNLIST
                VALUENAME "1601"
                ITEMLIST
                        NAME !!Disable VALUE NUMERIC 3
                        NAME !!Enable VALUE NUMERIC 0
                        NAME !!Prompt VALUE NUMERIC 1 DEFAULT
                END ITEMLIST
                END PART

                PART !!UserdataPersistence CHECKBOX
                VALUENAME "1606"
                VALUEON NUMERIC 0
                VALUEOFF NUMERIC 1
                END PART
        END POLICY

        POLICY !!Scripting
                PART !!ActiveScripting DROPDOWNLIST
                VALUENAME "1400"
                ITEMLIST
                        NAME !!Disable VALUE NUMERIC 3
                        NAME !!Enable VALUE NUMERIC 0 DEFAULT
                        NAME !!Prompt VALUE NUMERIC 1
                END ITEMLIST
                END PART

                PART !!AllowScriptPaste DROPDOWNLIST
                VALUENAME "1407"
                ITEMLIST
                        NAME !!Disable VALUE NUMERIC 3
                        NAME !!Enable VALUE NUMERIC 0 DEFAULT
                        NAME !!Prompt VALUE NUMERIC 1
                END ITEMLIST
                END PART

                PART !!ScriptJava DROPDOWNLIST
                VALUENAME "1402"
```

```
                ITEMLIST
                        NAME !!Disable VALUE NUMERIC 3 DEFAULT
                        NAME !!Enable VALUE NUMERIC 0
                        NAME !!Prompt VALUE NUMERIC 1
                END ITEMLIST
                END PART
        END POLICY

      POLICY !!UserAuth
                PART !!Logon DROPDOWNLIST
                VALUENAME "1C00"
                ITEMLIST
                        NAME !!AutoNTLM VALUE NUMERIC 0
                        NAME !!Prompt4Logon VALUE NUMERIC 65536
                        NAME !!Anonymous VALUE NUMERIC 196608
DEFAULT
                        NAME !!AutoIntranet VALUE NUMERIC 131072
                END ITEMLIST
                END PART
      END POLICY
END CATEGORY

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
CLASS MACHINE ;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

[strings]
Zone3Security="Internet Zone security Settings"
ActiveX="ActiveX Controls and plug-ins"
DownLoadSignedControls="Download signed ActiveX Controls"
AdminApproved="Administrator Approved"
Disable="Disable"
Enable="Enable"
Prompt="Prompt"
DownloadUnSignedControls="Download unsigned ActiveX controls"
ScriptUnsafe="Initialize and script ActiveX controls not marked
as safe"
InitAndScriptNotMarkedSafe="Initialize and script ActiveX
controls not marked as safe"
RunActiveXControls="Run ActiveX controls and plug-ins"
ScriptSafe="Script ActiveX controls marked safe for scripting"
Cookies="Cookies"
AllowPersistentCookies="Allow cookies that are stored on your
computer"
AllowSessionCookies="Allow per-session cookies (not stored)"
Downloads="Downloads"
FileDownload="File download"
FontDownload="Font download"
MicrosoftVM="Microsoft VM"
JavaPermissions="Java Permissions"
DisableJava="Disable Java"
HighSafety="High Safety"
LowSafety="Low safety"
MedSafety="Medium Safety"
```

```
Miscellaneous="Miscellaneous"
CrossDomainData="Access data sources across domains"
DontPrompt="Don't prompt for client certificate selection when
no certificates or only one certificate exists"
DragDropCopyPaste="Drag and drop or copy and paste files"
InstallDeskItems="Installation of desktop items"
LaunchProgsInIFRAME="Launching programs and files in an IFRAME"
SubFramesXDomains="Navigate sub-frames across different domains"
SoftwareChannelPerms="Software channel permissions"
SubmitNonencryptedForms="Submit nonencrypted form data"
UserdataPersistence="Userdata persistence"
Scripting="Scripting"
ActiveScripting="Active scripting"
AllowScriptPaste="Allow paste operations via script"
ScriptJava="Scripting of Java applets"
UserAuth="User Authentication"
Logon="Logon"
Anonymous="Anonymous logon"
AutoIntranet="Automatic logon only in Intranet zone"
AutoNTLM="Automatic logon with current username and password"
Prompt4Logon="Prompt for user name and password"

[IEAK]
Lock=1
Roles=111
NumOfDescLines=4
Platform=010
```