Global Information Assurance Certification Paper

Techniques for
Securing your Internet Explorer Browsing Experience at Home

Natalie M. Green
19 January 2004
GSEC Security Essentials
Version 1.4b, Option 1

Introduction

Browsing the Internet can easily introduce new vulnerabilities to a computer system that would not ordinarily be exposed. This can happen despite the presence of quality, properly configured antivirus software and a firewall. This paper endeavors to identify the risks of web browsing and suggest ways to mitigate them.

While some of these vulnerabilities are considered low-risk, annoying, or both, others can be downright dangerous. Low-risk annoyances can include homepage takeovers, cookies that track your browsing history, and advertisements that appear on your computer without your consent. The dangerous problems, however, can lead to hostile software (whether intentional or accidental) silently installing itself on your PC without your knowledge and actually *take over* your PC; the average computer user may not realize their system has been compromised. Even if they do, assessing and successfully repairing the damage can be difficult, if not impossible.

Target Environment

To address a typical configuration, this paper makes two assumptions regarding the user's computing environment:

1. Windows NT 4.0 with Service Pack 3 or higher is the operating system, though Windows 2000 or higher, with the latest Service Pack, is recommended. This is because: a) most of the computers connected to the Internet use NT 4.0 or higher; b) news regarding browser insecurities centers around Windows; and c) NT 4.0 was the first Microsoft OS to offer a good, readily manageable security model.
2. Internet Explorer (IE) version 5.5 or higher. There are three reasons for this: a) IE is very integrated into the Windows OS; b) it has become the most popular browser; and c) version 5.5 is the first that offers some of the security features discussed in this article, though IE6 with service pack 1 is recommended.
3. The target audience is the typical home computer user and small businesses, though many recommendations might also suit a large organization.

Roadmap

To mitigate the risks identified above, IE will first need to be configured to address as many of these deficiencies as possible, while not limiting its practical functionality. Next,

attempts will be made to close all remaining security holes by third-party software. To answer these problems, the proposed solutions will need to be: a) easily configured; and b) low cost or free so they will be accessible to most users. The potential problems, and their proposed solutions, are as follows:

1. Browser Configuration
2. Keep IE and Operating System Patches Updated
3. Cookie Management
4. Firewall Software and/or Hardware
5. Antivirus software
6. Sandboxing Mobile Code
7. Intelligent pop-up blocker
8. Ability to run browser with limited user rights/permissions

Two additional areas can be examined. However, they are typically reserved solely for the corporate environment, and will therefore not be explored:

1. Certificate Management; i.e., ability to control/accept certificates
2. Ability to run with a variety of cryptographic algorithms; for example, are there any IPSec, VPN, or SSL considerations?

Inspiration for Additional Internet Security

The initial idea for this paper came after my Windows 2000 laptop was compromised by a hostile webpage a few months ago. Though not a case study, it is worth providing a brief background on the discovery and removal process that was required, to give impetus for the importance of prevention.

ActiveX software was quietly and quickly installed on it without my consent or knowledge. Only by noticing extended network traffic did I suspect something was not right. Next, my CPU began spiking for longer than usual. I seriously suspected something awry had just taken place, and disconnected my laptop from the network so as not to: a) further infect it; and b) infect any other hosts on the network.

I then set about looking for what, if anything, happened. Being quite familiar with which folders should be where on my system, and knowing most of the file names and dates – particularly in the WINNT folder and below – I found several that did not belong. In the process of using Explorer's "Send To", I noticed that there was an extra entry with a bunch of foreign characters.

Each suspicious item was opened in Notepad, where additional files were found, as were references to new registry entries. These were opened, and all pointers to registry locations were examined. The same procedure was followed for all the miscreant files. Many "logical" files, folders, and registry entries were searched – with many of them yielding hidden content. All hostile material was saved to a zip file to record the evidence, and subsequently deleted.

This process of examination, combined with research on the Internet, revealed that the attack had manifested itself as an ActiveX control. I knew little about this type of mobile code – or any other mobile code, for that matter. How did it slip through? How did this code install itself without prompting me first? This security breach took place despite the fact that my laptop had the following industry-standard layers of protection:

1. Patches on the operating system and web browser were fully up to date;
2. It was behind a corporate hardware firewall;
3. Up-to-date top-rated antivirus, firewall, anti-worm and anti-trojan software were properly installed and configured; and
4. A well-rated pop-up blocker further protected it.

Clearly, some security features were missing and/or not properly configured on my laptop in order for this breach to have occurred.

It took several hours to find and remove all traces of the attack, which bound itself to many running processes, thus making its detection and deletion difficult. The cleaning process required specialized software and intimate knowledge of many aspects of the operating system and software on it; this included files and registry entries. The typical home user should not be expected to possess the requisite knowledge of how to protect or configure against these problems. These tools and skills are simply not readily available to the typical PC user – home, corporate, or otherwise.

I immediately realized that this could easily happen to many innocent home users, and they would have no way to discover it – let alone fix it. I sought to devise a simple, effective solution that costs little or no money. They are listed and addressed here – one by one.

Steps to Surfing Safely

1. Browser Configuration

Some improvements to IE's security can be made by simply making a few adjustments, though some caution should be observed. Specifically, a lot of "content" on the web (e.g. sounds, advanced display methods, and video)[1] depends on the very technologies that are exploited by others. Therefore, where IE offers only limited configuration options – either On or Off – software will be suggested in later steps to mitigate these deficiencies.

The following configuration for IE is recommended, using its configuration tabs as reference:

a. "General" tab: click "Settings…" button, then change "Check for newer versions of stored pages" to "Every visit to the page" so as to avoid viewing stale content, and change the "Amount of disk space to use" to about 50 MB. This will

[1] Freedman, Alan, Computer Desktop Encyclopedia, 9th Edition, Berkeley: Osborne/McGraw-Hill, 24 September 2001

be more than adequate for even the most graphics-intense websites, and limits your disk space usage to a manageable level for storing temporary files.

b. "Security" tab: Set "Internet Zones" and "Restricted Zones" to "default"; these are "medium" and "High", respectively.
c. "Privacy" tab: IE does not provide management of cookies, just adherence to a "privacy policy", which should be "Medium" (the default setting), in order to comply with recommendations made by the cookie software mentioned later in this article.
d. "Content" tab: IE's default setting is to store usernames and passwords; while some regard this as a "feature", this information can be retrieved by malicious users over the Internet, thus posing serious potential security threats to accounts like email. Click "AutoComplete…" and uncheck "User names and passwords on forms", making sure that "Prompt me to save passwords" is also unchecked.
e. "Connections" tab: Make sure the radio button labeled "Never dial a connection" is selected so as to prevent spyware from spontaneously dialing "home" with information about you and/or your surfing habits.
f. "Programs" tab: no setting changes needed, except possibly changing the default "HTML editor" to "Notepad", so as to avoid accidentally running embedded web code when opening such a file in Microsoft Word, or some other powerful web editor.
g. "Advanced" tab: The following security settings should be enabled: "Do not save encrypted pages to disk" and "Empty temporary Internet Files folder when browser is closed". Additional restrictions can be investigated and implemented, but these two provide protection against remote retrieval of potentially sensitive information stored in your browser's temporary file folder.

2. Keep IE and Operating System Patches Updated

By going to Microsoft's update website, http://windowsupdate.microsoft.com, and applying the available patches, users can ensure that Internet Explorer and their operating system are protected against the posted vulnerabilities. This can be done by using any or all of the following options: a) Windows' "Automatic Update" feature, which was included in WinXP and was introduced to Win2k by SP4; or b) manually going to the http://windowsupdate.microsoft.com website on a regular basis (e.g., about weekly). It is worth mentioning that corporate users have additional means of performing these tasks, like Microsoft's SMS or SUS ("Software Update Services").

Important Note to Microsoft Outlook and Outlook Express users:
These products use Internet Explorer as the engine to render HTML-formatted email. This makes these products capable of running active code, such as ActiveX and Java, just like Internet Explorer.

One more suggestion regarding Outlook: make sure it is using all the latest security packs and patches; these updates are *not* included in the usual WindowsUpdate mentioned above (although Outlook Express's updates are included); rather, they must be obtained from Microsoft at http://office.microsoft.com/officeupdate/default.aspx. You

will need the original Microsoft Office CD(s) during this procedure. Repeat this process until all patches and configuration to IE and Microsoft Office will have been applied, and the operating system patched, since some updates are dependent on others having been installed first. Remaining vulnerabilities will have to be addressed with third-party software.

3. Cookie Management

Cookies are small text files that are placed on your computer to identify you to the website that placed it (or them), and include some personal information about your browsing session[2]. Typically, this can be a good thing.

For example, suppose you go to the website of your favorite clothing company. They may place a cookie on your computer which identifies which items you looked at on their site and, using that information, make guesses as to what kind of clothes you might be interested in, then present those to you when you browse to their site again. Cookies are also used for keeping security session information, like when you go to your bank's website: they'll give you a cookie that contains encrypted information regarding your banking session.

However, companies can use cookies to track your browsing habits and organizations that you may *not* want to track you. For example, a company may place a cookie on your computer, then use it to track which websites you visit and how often. A specific example can be found on DoubleClick.com[3]. While this information in and of itself is not a direct security violation (i.e., they have not done anything else to your computer but place and read a cookie), this information can be used to harvest information that you consider private.[4]

To that end, this is really about privacy management, and many people are understandably concerned. IE5.5 and higher include ways to handle cookies better than their predecessors, the configuration of which was addressed above. Much in the way of its built-in cookie management, however, is lacking. Here's where additional software comes in.

"Cookie Crusher" is US$15 and can turn many cookies from needlessly long expiration periods (20 years or more!) to "per session", where the cookie is deleted once you close the browser. Cookie tracking can be essentially eradicated with this software.

Some of its features include: a) ability to block cookies created by JavaScript; b) ability to block "referrer" cookies[2]; c) white- and black-list management; and d) ability to work with any browser, such as Netscape, due to the fact that it functions as a proxy to the

---

[2] Freedman, Alan, Computer Desktop Encyclopedia, 9th Edition, Berkeley: Osborne/McGraw-Hill, 24 September 2001

[3] Olsen, Stefanie; and Kane, Margaret. "'Reassurance' a key word as Google grows." CNET News.com. 03 March 2003. URL: http://news.com.com/2100-1032-990685.html (14 January 2004)

[4] Stein, Lincoln D. "Q10: Do Cookies Pose any Security Risks?" 23 February 2003. World Wide Web Consortium. URL: http://www.w3.org/Security/Faq/wwwsf2.html#CLT-Q10 (14 January 2004)

Internet connection. Note that Cookie Crusher requires the installation of Microsoft Windows' .NET software, which is available freely from Microsoft's WindowsUpdate website.

4. Firewall

This is one of two components that you do *not* want to do without; antivirus software is the other, and is discussed in the next Step. It is virtually required for any notion of security when connected to a network, regardless of whether or not the host machine is used to surf the Internet.

There are two types of firewall available: hardware and software. Optimally, both will be used. That approach is sometimes used in corporate environments as part of a defense-in-depth strategy, but that solution can add significant cost and complexity. For a typical home with one or two PCs, I recommend a software implementation.

My choice is Agnitum's Outpost Pro 2.0. It is "extensible", meaning that its abilities can be extended by the use of plug-ins. This provides the distinct benefit that anyone can create their own plug-in, like a content-filter. In addition to the typical benefits afforded by software-based firewall, the US$40 Outpost includes:

a. Optional password protection, thus preventing the software and its configuration from being changed by accident – or deliberately;
b. Extensive logging abilities, giving you the ability to see what's happening "under the hood";
c. Tools and methods to deploy the software and its configuration with ease throughout an organization;
d. Stateful packet inspection; and
e. Highly customizable rulesets.

5. Antivirus software

Though good antivirus software did not save my laptop from the attack mentioned in the beginning of this article, it *has* protected my laptop from several nasty programs, and has done the same for millions of other computers worldwide for many years. In other words, this is a critical piece of software that should be installed on virtually all computers.

Personally, I use Trend Micro's corporate antivirus software, OfficeScan. Setup is quite straightforward, and they provide updated signatures weekly, with incremental updates available to address emergencies, testing, or even routine deployment in a corporate environment. It also provides protection for POP3-based email, in addition to the typical SMTP-based email. Given that most PCs at home will use POP3, this is an important feature. Another mode of protection is for PDA file transfers. PDAs can accept and transmit files, thus making it possible for them to propagate viruses. With Trend, you can be protected against such possible avenues of attack.

Trend's home antivirus software, PC-Cillin, costs US$50, includes the above features, plus anti-spam and firewall software. When I needed technical assistance, their

representatives were prompt and friendly, both via phone and email. Regardless of which vendor is chosen, it is recommend that the user runs a quality, up-to-date antivirus software.

6. Sandboxing Mobile Code

"Mobile Code," (also called "controls") is defined as "code that can be transmitted across the network and executed on the other end"[5]; examples of such code are ActiveX, JavaScript, Java, and Visual Basic Script (VBS), and the "other end" is your web browser. These types of controls are typically used to provide you with the benefits of an enhanced, rich website.

Without restrictions, mobile code could act just like any other program on a computer. Since this approach is *usually* more than what's required (e.g., a training demo, or Tech Support performing remote control of a user's PC), Sun implemented what's called a "sandbox" around their Java code. Their sandbox imposes the following restrictions on what Java code can do[6]:

- Read from a file system on the client machine
- Write to a file or delete it
- Delete a file on the file system
- Connect to a network port on any machine other than the HTTP server it came from
- Execute another program, load a library or DLL, and so on.

Implementations of this "sandbox" vary from engine to engine, but they aim to limit the extent and abilities of the code. For the most part, Java does this well. However, one type of software that deserves special attention is Microsoft's ActiveX technology, because it does not use a sandbox. The only security control afforded this tool is whether the developer has "signed" his or her code. The process of signing code, termed "authenticode" by Microsoft, is described by Freedman as "a method of ensuring that an executable program is coming from a valid software publisher".[7]

What does this mean for you, the end-user?

a. You – the end-user – must decide if *you* trust the code the developer wants to install on your system; and
b. You must be able to figure out if the signature on the code actually belongs to the person or group it claims. In other words, just because it says "Signed by Acme", *anyone* can actually call themselves Acme and subsequently sign the code.

This security model requires a great deal of trust on your part, the Internet surfer. Why was it designed this way? So as to greatly enhance and extend the abilities of the

---

[5] Conolly, Dan. "Mobile Code." 12 September 1996. URL: http://www.w3.org/MobileCode/ (14 January 2004)
[6] Nanda, Nitin; Kumar, Sunil. "Breaking the Sandbox Barrier, Part 1." Jupitermdia Corporation. 04 February 2001. URL: http://www.developer.com/java/article.php/934031 (14 January 2004)
[7] Freedman, Alan, Computer Desktop Encyclopedia, 9th Edition, Berkeley: Osborne/McGraw-Hill, 24 September 2001

browser. For example, let's say you've gone to a business associate's website and accepted their developer's ActiveX code, which creates a button enabling you to easily print to their printer, Printer1. Upon visiting the website weeks later, the printer name has been changed. Since you decided to trust that ActiveX code from the developer of that website, they can automatically update their code on your browser with the updated name of the printer, now Printer2.

This "feature" of unrestricted power was designed to enable enhanced "content", like "web pages, images, music, audio, [and] driver and software downloads"[7]. ActiveX controls, however, can be harmful, like "Internet Exploder" developed by Fred McLain[8]. A particularly insidious example would be for a malicious website to remotely run buggy, insecure code that had previously been downloaded by a user, and then exploit it for their own purpose[9,10]. In Microsoft's own words, "An ActiveX control can be an extremely insecure way to provide a feature. Because it is a Component Object Model (COM) object, it can do anything the user can do from that computer"[11].

Microsoft has attempted to overcome this vulnerability by implementing a process they call Authenticode, which allows the developer to obtain a certificate from a trusted authority, like Verisign, and subsequently sign their code with the certificate; in so doing they are legally stating that their code is not malicious. However, this is thought by some to be a poor security model[12]. There are several ways of mitigating these potential risks:

a. Block their access to your system altogether. While this would work, it would drastically limit your ability to enjoy much of the web's content, like streaming media, some types of online transactions, etc.
b. Another is to filter them using a known list of "bad" code. This requires the use of a database, which must be constantly updated. The problem with this approach is that someone must have already found and reported the dangerous code to the list you decide to use, and your database must be up to date. In other words, all a "bad" person has to do is alter existing "bad" code to make sure it isn't flagged against the database, or simply create nasty new code.
c. Prompt you with a warning each time a script attempts to run, and offer you the option of discarding it or running it. While sometimes helpful, this approach it can quickly be cumbersome: it places the question of trustworthy code squarely on the shoulders of the end-user many times – most of them not necessary.

---

[8] McLain, Fred. "The Exploder Control Frequently Asked Questions." February 1997. URL: http://www.halcyon.com/mclain/ActiveX/Exploder/FAQ.htm (14 January 2004)
[9] CERT. "Results of the Security in ActiveX Workshop." CERT Coordination Center. 03 January 2001 URL: http://www.cert.org/reports/activeXreport.pdf (14 January 2004)
[10] Grimes, Roger. "Malicious Mobile Code." O'Reilly. August 2001. URL: http://www.oreilly.com/catalog/malmobcode/chapter/ch11.html (14 January 2004)
[11] Microsoft. "Designing Secure ActiveX Controls." Microsoft Corporation. URL: http://msdn.microsoft.com/library/default.asp?url=/workshop/components/activex/security.asp (14 January 2004)
[12] Fleishman, Eric. "Code Signing." IPJ Volume 5, Issue 1. March 2002. URL: http://www.okena.com/en/US/about/ac123/ac147/ac174/ac177/about_cisco_ipj_archive_article09186a00800c83db.html (14 January 2004)

d. Use software that intercepts each applet as it's downloaded, allowing each mobile code to run in its own "sandbox". Only when the applet attempts to break out of the sandbox would you be provided the details of the questionable code's actions and asked whether or not to permit it. This provides a much greater mode of protection, but there's one catch: you have to trust the software that provides this protection.

I recommend the forth choice – option d. One company that attempts to use "sandboxing" technology is Finjan software. For US$30, the home user can choose their SurfinGuard Pro. It's easy to install and the default setting provides excellent coverage. One feature is its "white list", which prevents the software from monitoring anything on the link or site you enter. This can be useful for sites like Microsoft's WindowsUpdate, which needs ActiveX to run with untethered, un-"sandboxed" Administrator privileges on your PC. Unfortunately, however, this "white list" cannot be viewed or edited, but merely cleared, so it is recommended to maintain a separate file of all sites added to the list.

Until SurfinGuard Pro's functionality is enhanced to remedy this drawback, corporate users will likely want to choose Finjan's SurfinShield, which provides a high degree of granular control of this particular part of the program. It is worth noting that Cisco now uses Finjan's sandboxing technology, SurfinGate, in their PIX firewalls[13].

7. Running the Browser with Limited Security Credentials

One reason my laptop was infected that night was because I was logged in as an administrator, which has full abilities to run any software on the system. Most users will install their system in this mode by default. Therefore, since I was running my browser in the same user context, the hostile code downloaded by the nasty website I visited silently installed its code with the same permissions: administrator.

Most home computers running Windows NT or higher with IE 5.5 are vulnerable to the same problem: they log in as the default user – administrator – and proceed to do all web browsing with the same account, thus opening their system to similar attacks.

So why don't we just have all users log in with either 'User' (which has very limited permissions by default), or 'Power User', which has additional rights, but not as many as administrator? This would involve a great deal of user education, which is at odds with one of the foundations of this paper: to make it easy for the average home user to implement these solutions.

There is an built-in alternative: use the "run-as" service to have the typical user log in as usual (i.e., "administrator"), and run Internet Explorer with lower credentials. While this would work, it requires a command-line interface, which must be either typed each time it's run or put into a batch script – which must contain the restricted-user's password in

[13] CBR Online. "Roaming Bandits." ComputerWire. 01 February 1999. URL: http://www.cbronline.com/research_centres/7f633cde3e499d0080256d350047d61f (14 January, 2004)

clear text – not a good idea. Creating a batch script is definitely beyond the ability of most users.

There is yet another choice: for US$14, you can buy "RunAs Professional" that provides a GUI and stores the account password using strong, 256-bit AES encryption. It's easy to set up and use, and gives the user a secure means of running Internet Explorer – or any other software on their PC, for that matter. RunAs Pro is available from http://www.mast-computer.com/c_9-l_en.html

This is the approach recommended for *all* users. It is my belief that, after antivirus and firewall protection, this product provides the best security for your computer. It can protect numerous programs and is easy to run.

In order to successfully use this software and those like it, however, the user *must* first follow these steps in order:

a. Create the restricted-user account (this is done from within "Local Users and Groups");
b. Log out of their current computer account and log in to the one they created;
c. Open Internet Explorer; this will automatically walk the user through the process of configuring the settings to surf the web;
d. Apply the steps made in Part 1 of this paper; and then
e. Log out of the restricted account and log in with the "usual" account (e.g., "administrator").

This process should be repeated for any programs to be used with RunAs Pro (except those in "d.", since they apply only to Internet Explorer). If they are not followed, the user will usually find that some or all of the program's functionality will be unavailable. This caveat aside, RunAs Pro provides an easy icon for the user to click when wanting to run Internet Explorer in a secure fashion.

8. Pop-Up Blocking Software

The risk of pop-ups is the same as that of going to site that contains malware; for example, the pop-up can redirect you to hostile sites. Pop-ups can also have the same effect as a Denial of Service attack (DoS), in that they can bombard, and thus disrupt, your work environment. This could be done by either: a) changing the focus of where you are typing; b) utilizing most of your computer's processing power; or c) causing your computer to crash after all its resources have been exhausted in displaying the pop-ups.

Virtually everyone who has surfed the web has experienced pop-ups. While some of them are annoying or even harmful, others are helpful and required. This is due to the fact that there are different *types* of pop-ups: good and bad.

An example of a "good" pop-up could be when you are purchasing a product on the web and click the "Confirm" button to complete the transaction, a window may appear, asking you to wait while your credit card is authorized.

I have settled on one software called “Popup Block”, available at http://www.popupblock.net. For US$15 it offers protection from all but one type of “bad” pop-up – creation of a batch-script to launch a pop-up – found at testing sites[14,15].

While there are many free pop-up stoppers out there, Popup Block was the only one found by this author to pass all but one of the tests. That one failure is the execution of batch scripts *created* by the pop-up; this author has never found an instance of that type in the "wild", and thus does not consider it to be serious at the present time.

Summary

The following steps are recommended:

a. Implement Step 1 from “Steps to Surfing Safely” to tighten Internet Explorer as much as possible without reducing its ability to correctly render webpages. If there are any “special” websites that are used, like a company Extranet, then it is advised that they be tested at this point. If they test out, then it’s safe to move on to the next step.
b. Ensure all applicable patches are downloaded and applied.
c. If enhanced privacy is desired, then download, install, and test out cookie management software. I recommend Cookie Crusher 3.2, but there are two caveats: it requires Microsoft’s .NET software and its configuration takes a bit of fine-tuning to get it to work best for your environment. I suggest trying it. Once you get it set the way you want, it works very well –even for other browsers, like Netscape.
d. Choose a firewall and install it, unless you’re happy with the one(s) you’re already using. I prefer Agnitum’s Outpost Pro 2.0, a software solution. So long as you have a quality firewall, whether it be hardware and/or software-based, is wholly up to you.
e. Choose a good antivirus vendor, and install their software. Trend Micro is my choice for several reasons – for either the home or corporate user. For the home user, their PC-Cillin antivirus software includes firewall and antispam capabilities.
f. I would give “sandboxing” software serious consideration. Finjan’s home product SurfinGuard Pro 5.7 is only $30, provides very good protection, and the only configuration required is adding sites to its “whitelist” that you don’t want to be sandboxed (like WindowsUpdate). I believe that, as the months and years go by, software like this will give you piece of mind from increasingly hostile websites.
g. Running Internet Explorer with limited rights is a very safe move, and only requires the user to follow a few easy steps of configuration. RunAs Pro, at $14, gives the user a quick, easy way to run their browser, while protecting the password to the limited-user account with very strong encryption.

[14] Gransee, Marco. “Test your popup blocker software.” [No date listed]. URL: http://www.popuptest.com/ (14 January 2004)
[15] Kaul, Sergei. “Tests.” [No date listed]. URL: http://www.popup-killer-review.com/test.htm (14 January 2004)

By taking these steps, users can experience enhanced privacy and very good protection for their computer from malicious programs while surfing the web with Internet Explorer. With these protections in place, my laptop would have been fully protected against the hostile ActiveX code that infected my machine during the summer of 2003. Moreover, I believe that if the home user had these protections in place and accidentally visited the malicious website, their computer would also have been protected.

References

CBR Online. “Roaming Bandits.” ComputerWire. 01 February 1999. URL: http://www.cbronline.com/research_centres/7f633cde3e499d0080256d350047d61f (14 January, 2004)

CERT. “Results of the Security in ActiveX Workshop.” CERT Coordination Center. 03 January 2001 URL: http://www.cert.org/reports/activeXreport.pdf (14 January 2004)

Conolly, Dan. “Mobile Code.” 12 September 1996. URL: http://www.w3.org/MobileCode/ (14 January 2004)

Fleishman, Eric. “Code Signing.” IPJ Volume 5, Issue 1. March 2002. URL: http://www.okena.com/en/US/about/ac123/ac147/ac174/ac177/about_cisco_ipj_archive_article09186a00800c83db.html (14 January 2004)

Freedman, Alan, Computer Desktop Encyclopedia, 9th Edition, Berkeley: Osborne/McGraw-Hill, 24 September 2001

Gransee, Marco. “Test your popup blocker software.” [No date listed]. URL: http://www.popuptest.com/ (14 January 2004)

Grimes, Roger. “Malicious Mobile Code.” O’Reilly. August 2001. URL: http://www.oreilly.com/catalog/malmobcode/chapter/ch11.html (14 January 2004)

Kaul, Sergei. “Tests.” [No date listed]. URL: http://www.popup-killer-review.com/test.htm (14 January 2004)

McLain, Fred. “The Exploder Control Frequently Asked Questions.” February 1997. URL: http://www.halcyon.com/mclain/ActiveX/Exploder/FAQ.htm (14 January 2004)

Microsoft. “Designing Secure ActiveX Controls.” Microsoft Corporation. URL: http://msdn.microsoft.com/library/default.asp?url=/workshop/components/activex/security.asp (14 January 2004)

Nanda, Nitin; Kumar, Sunil. “Breaking the Sandbox Barrier, Part 1.” Jupitermdia Corporation. 04 February 2001. URL: http://www.developer.com/java/article.php/934031 (14 January 2004)

Olsen, Stefanie; Kane, Margaret. “’Reassurance’ a key word as Google grows.” CNET News.com. 03 March 2003. URL: http://news.com.com/2100-1032-990685.html (14 January 2004)

Stein, Lincoln D. “Q10: Do Cookies Pose any Security Risks?” 23 February 2003. World Wide Web Consortium. URL: http://www.w3.org/Security/Faq/wwwsf2.html#CLT-Q10 (14 January 2004)