Interested in learning more about security?

SANS Institute InfoSec Reading Room

Securing Internet Explorer Through Patch Management

Internet Explorer (IE) is often overlooked when it comes to the defense-in-depth of a corporate network. Not keeping IE updated with the latest security patches can leave your entire system vulnerable to malicious hackers. Resources and money are tight in the Information Technology world, and trying to stay on top of the amount of patches that are released each year is a rigorous task. Within this paper I will show you the importance of keeping IE patched throughout your network. I will go into the current state of pat...

Securing Internet Explorer Through Patch Management

Student Name: Benjamin Meader
Date Submitted: August 4, 2003
Assignment Version: GSEC 1.4b

Table Of Contents

Abstract....2
Introduction....2
Patching: How Hard Can It Be?....3
What Hackers Can Do To an Insecure Browser....5
Patch Management For a Secure Browser....9
Establish Strong Security Settings....10
How to Keep IE Current With Recent Security Patches....14
Conclusion....17

Abstract

Internet Explorer (IE) is often overlooked when it comes to the defense-in-depth of a corporate network. Not keeping IE updated with the latest security patches can leave your entire system vulnerable to malicious hackers. Resources and money are tight in the Information Technology world, and trying to stay on top of the amount of patches that are released each year is a rigorous task.

Within this paper I will show you the importance of keeping IE patched throughout your network. I will go into the current state of patch management and demonstrate what could happen to your network if you leave IE unpatched. You will also receive inside information on how to mitigate the risk of IE being attacked through the application of strong security settings. Finally, I will discuss the need of a repeatable methodology for testing and deploying patches and how important it is to establish one at your firm.

Introduction

Internet Explorer (IE) is the world's most prolific browser. By the end of 2002, IE was the browser of choice on 95% of the world's computers that were accessing the Internet.[1] IE is included in the latest version of Windows XP and has been included in Microsoft's operating systems for the past eight years[2], allowing it to dominate the browser market. It is this saturation that has helped IE become a target for malicious hackers.

At least once every month or two, Microsoft releases yet another IE Cumulative Security patch which is deemed critical to the overall security of your systems. In firms both small and large, IE weaknesses are often overlooked as vital to the overall defense-in-depth of their networks. No matter how secure your firm's firewall may be, an unpatched browser can leave systems exposed and make them easy targets for hackers. It is imperative that your firm keep its systems patched or it may run the risk of leaving itself open to some serious attacks.

Deploying patches is not as simple as it sounds. Resource constraints, budget concerns, and the fear of a patch not working have contributed to a laissez faire attitude towards keeping IE patched. A general lack of money and resources keeps most IT shops from realizing the importance of establishing and maintaining a consistent methodology for testing and deploying patches for IE. An untested patch could adversely affect the functionality of a firm's web application.

To mitigate potential hacking threats, the best way to protect your firm's system is to maintain strong security settings within a securely patched browser. IE is so closely tied to the operating system that keeping it updated with the most current security patches is paramount to maintaining a secure network.

Patching: How Hard Can It Be?

Security experts estimate that barely 50 percent of all software security patches are applied by enterprise IT administrators.[3] It is difficult to obtain actual percentages due to the fact that large enterprises often download a patch to a local server then redistribute it. However, it is obvious that patching is not being done nearly as much as it should be.[4] IE is often overlooked as administrators from large firms mistakenly believe that their systems are safe as long as a firewall is in place. In "When Patches Aren't Applied" by John Naraine, Thomas Kristensen from the security firm Secunia states:

> Sometimes, they will hesitate and delay fixing a faulty browser for several months and assume they aren't vulnerable because they're using a firewall but that is a dangerous assumption. The intruders are sophisticated and are using attack scenarios that penetrate the firewall.[5]

Why is patching not being done? Administrators who are aware of the need to keep IE secure face a tremendous challenge. Each time a patch is released, they must take several factors into account before considering deploying it such as:

- Does this patch address a critical flaw that is important to the overall network security?
- Will this patch disrupt any web applications currently used at my firm?
- Are time and resources available to adequately test patches and maintain new patch releases?

Too Many Patches and Too Few Resources

In larger firms especially, installing patches can be quite an undertaking. Enterprise IT administrators have many tasks and keeping the browser patched with the latest security updates is not always their highest priority.

With the large amount of patches that are released each year it makes it difficult to stay on top of the workload. The volume of security warnings and patches that are available (Microsoft released a whopping 72 security related bulletins in 2002) can be overwhelming to administrators.[6] While not all of the bulletins and warnings released by Microsoft are related to IE, an IT department faces a tremendous task as it struggles to keep up with all Microsoft product patches. According to The CERT Coordination Center (www.cert.org), more than 4,000 vulnerabilities were reported last year. In just the first quarter of this year, more than 900 vulnerabilities have been reported.[7] "Too many vulnerable systems are left in exposed situations, even when patches have been made available."[8]

Well-known exploits that have been around for a long time are still being used because administrators have a difficult time keeping up with the amount of patches which need to be applied. "Most IT organizations fall down in trying to keep up with the overload of alerts, patches and upgrades. However, the majority of attacks come from known vulnerabilities with available patches, such as the SQL Slammer worm."[9]

The amount of testing required to make sure that a patch doesn't do more harm than good is a tremendous expenditure and needs to be mitigated. The current lack of resources within many IT departments may not perform the adequate testing needed before the patches are deployed within a firm. Without proper testing, you run the risk of deploying a patch that is virtually impossible to remove without reinstalling IE. Many firms believe that this expenditure in time and resources is not warranted. However, it only takes one successful attack to change their minds.

More Harm Than Good?

With patch deployment, there is always the possibility that the patch will do more harm than good. If the patch is deemed critical to the security of the network, it needs to be deployed. When systems are running smoothly, it is hard to justify the need to patch a potential hole when it could disrupt the continuity of a web application. In a large firm with hundreds of web applications this can be a major hurdle.

The reason patches do not always work flawlessly is due to the pressure that developers who create the patches are under. The development of a patch begins with finding an exploit, usually by a responsible developer who makes Microsoft aware of the problem. Microsoft is then under the gun to develop a fix for the hole and release the fix as soon as possible. Many times, it can take months for Microsoft to release a patch. Because of the pressure to release the patch, there have been cases where the patch did not work properly. As recently as May of 2002 a critical security patch released by Microsoft did not adequately address the problems it was supposed to patch. GreyMagic Software, an Israeli security advisory company, felt that Microsoft did not correctly solve the six vulnerabilities listed within its security bulletin, "They only patched a symptom of this vulnerability, not its root cause," the posting said, "As a result of that incomplete patch, IE5 and IE5.5 are still very much vulnerable to this attack in other resources."[10] Sometimes, revisions to the patches themselves are released to fix a problem that was caused by the installation of the patch. It is no wonder that many firms do not patch their systems in a timely manner.

Even if your firm is uncertain that the patch will work, it is usually a best practice to maintain updated patches on your system. As IT administrators, we have to trust Microsoft not to lead us astray on the patch's functionality. However, Microsoft does not guarantee that their patches will not adversely affect a home-grown application and it is up to you to make sure that those applications will continue to run smoothly even after a patch is implemented. Not applying a critical patch can lead to a host of problems. "The consequences may include disclosure of cookies, local files or data, execution of local programs, download and execution of arbitrary code or complete takeover of the vulnerable system."[11]

It is vital that we keep our systems updated to keep the threat of a hacker who may obtain sensitive information contained on the network to a minimum.

What Hackers Can Do To an Insecure Browser

There are many different ways for a hacker to gain entry to your network through an insecure browser. One of the more dangerous means of access is Cross Site Scripting, otherwise known as XSS. Due to the growing number of dynamic websites that allow sites to deliver customized output, XSS has become a favorite way for hackers to circumvent even the most robust of firewalls by attacking the browser itself. It requires someone from within the network to visit a website or message board outside of the firewall and activate a script unbeknownst to the user. This script is usually injected in the form of JavaScript, VBScript, ActiveX, HTML, or Flash.[12] "Everything from account hijacking, changing of user settings, cookie theft/poisoning, or false advertising is possible."[13]

Many ways of exploiting IE so that these scripts can run have been found. A recent Microsoft security bulletin (MS03-004) released in February of 2003 patches an XSS exploit within IE.[14] This particular exploit uses the showHelp function within Windows. This function is normally used to display help files.

Example of showHelp Exploit

The showHelp hole could allow an attacker to access user information, invoke executables already present on a user's local system, or load malicious code onto a user's local system.[15] This exploit was discovered by Andreas Sandblad on October 30, 2002 and was patched by Microsoft on February 5, 2003.[16] Sandblad runs a web site that outlines many of the vulnerabilities the browser may have.[17] There are various other types of exploits that can be used to manipulate IE. Cross site/zone scripting errors are some of the more dangerous vulnerabilities that can be exploited on the browser.

This particular exploit points out the problem in IE with the showHelp function. showHelp is designed to show help files that have the .chm extension. Sandblad discovered a hole within showHelp that can "fool" IE into thinking it is in the Local Computer Zone, which usually has very low security.

> Basically, there are several security restrictions on the url argument:
> 1. Only urls starting with "file:" or "http:" are allowed.
> 2. If the url points to a local resource, then it must be a compiled help file with file extension .chm.
> 3. Compiled help files downloaded using the http protocol are not trusted.
>
> The problem is that if you call showHelp with the argument "file:" then security restriction (1) gets disabled.

> So, what can be achieved with (1) disabled? One way is to take advantage of the JavaScript protocol. Remember that JavaScript code in a url with the JavaScript protocol will be operating in the same site/zone as the url it is applied over.
>
> Examine the following JavaScript code:
>
> showHelp("file:") //Disable security restriction
> showHelp(url); // Let url be in another site/zone
> showHelp("javascript:"+code); // Operating in the same site/zone as url [18]

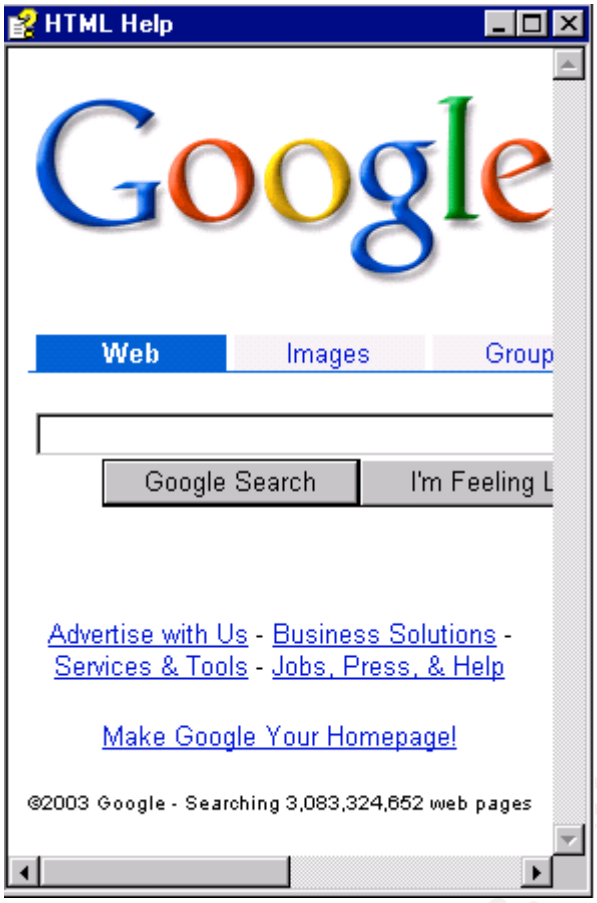
Sandblad goes on to demonstrate the ease with which an attacker could insert JavaScript to find very useful information or even run an executable on the local system. The four examples he demonstrates are:

1. Read arbitrary cookies
2. Read any local file using the XMLHTTP ActiveX control
3. Read viewable local files the old way
4. Using the "execute programs with parameters" method to run programs on a victim's system.[19]

As you can see from these examples, this is a very serious hole that needs to be patched as soon as possible. Let's take a closer look at the first example and see how it works on an unpatched system. In this exploit, Sandblad shows us how to read an arbitrary cookie. I customized his code in the example and inserted it into an HTML file I named Sandblad1.Html:

```
<SCRIPT>
showHelp("file:");
showHelp("http://www.google.com/")
showHelp("javascript:alert(document.cookie)"
</SCRIPT>
```

Here is what happens when I run that particular script on an unpatched browser. The first thing that happens is that the HTML Help window pops up with the Google website displayed in it (see following image).

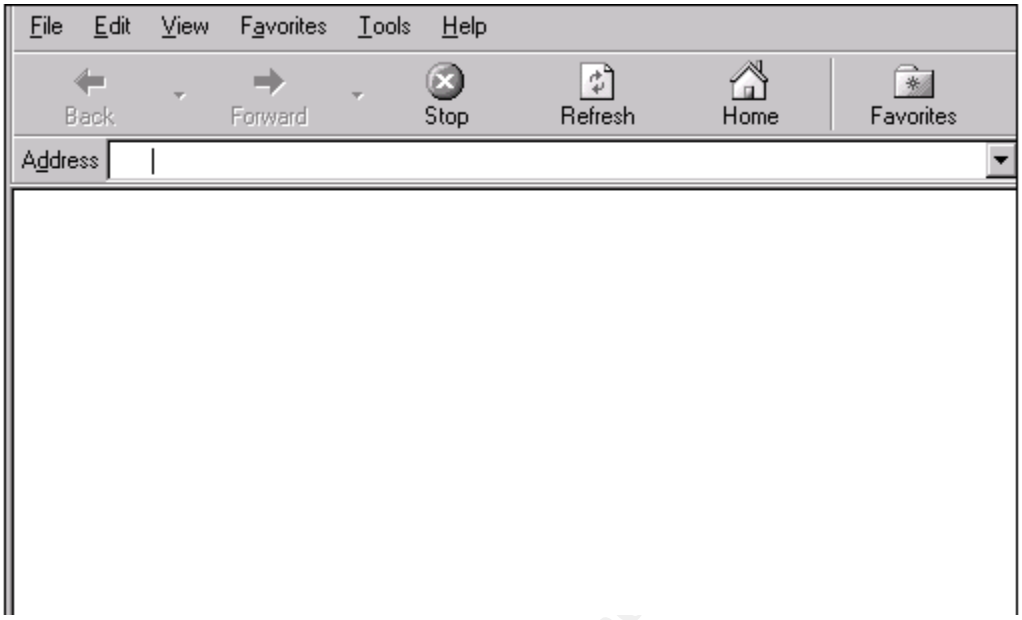


At the same time, in a separate window, the script displays the cookie information for Google (see following image).

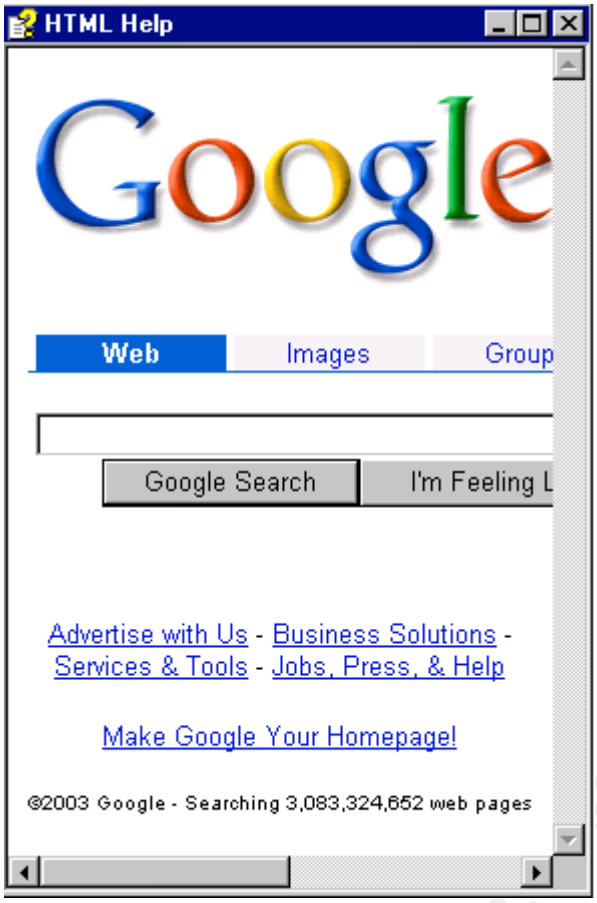
We are clearly dealing with a cross site/zone scripting issue.[20]

By executing this HTML file I can read the cookie file for Google. This means that as a malicious web site operator, I could simply attract someone to my site, or have them run the script through an e-mail attachment. As long as I know the url I need to access, I could trap the user into executing this script, which would give me the cookie information. This is a clear case of an XSS exploit that needs to be patched.

After applying the patch to this system, the browser acts as it should and does not allow the script to access to the HTML Help, instead a blank screen appears (see following image).



The cumulative patch has disabled HTML Help altogether. So, while this fixes the current problem, it also adds another problem by making the showHelp function unavailable for other applications. Microsoft then released a separate hotfix so that HTML Help would be back, this time without the hole. This was the HTML Help 1.40 Update patch which was released on February 11, 2003.[21] After applying this patch and running Sandblad1.html, the HTML Help function works as it should. This time, after running Sandblad1.html, the showHelp function does not display the cookie information in a separate window. It does, however display the Google site in the HTML Help window (see following image).



The exploit above demonstrates three important things about Internet Explorer:

- It does not take sophisticated code to cause a lot of harm.
- Patch testing is a must—any patch could disable functionality to another web application.
- It takes time for Microsoft to develop and release a patch for exposed vulnerabilities—in this particular instance it took almost four months.

Clearly, we need to protect our systems from these types of attacks. The best way would seem to be by applying the patch as soon as it comes out, which is recommended by Microsoft. However, this is also an example of a patch taking away a certain feature in order to protect the machine. The patch disabled the HTML Help functionality that is used to display help (.chm) files. By disabling this, you run the risk that help files would no longer display on any of your web applications. This is an example of why certification testing is needed.

Patch Management For a Secure Browser

The vulnerability above highlights the tremendous insecurity that faces network administrators in trying to keep IE secure. There are three main steps to securing the browser at your firm:

- Establish strong security settings.
- Test recent patches against your applications.
- Deploy security patches in a timely manner.

Establish Strong Security Settings

Security within IE is broken down into four distinct Security Zones which are the:

- Internet Zone
- Local Intranet Zone
- Trusted Sites Zone
- Restricted Sites Zone

Each zone contains specific security settings that you need to be aware of to make sure that IE is set up correctly.

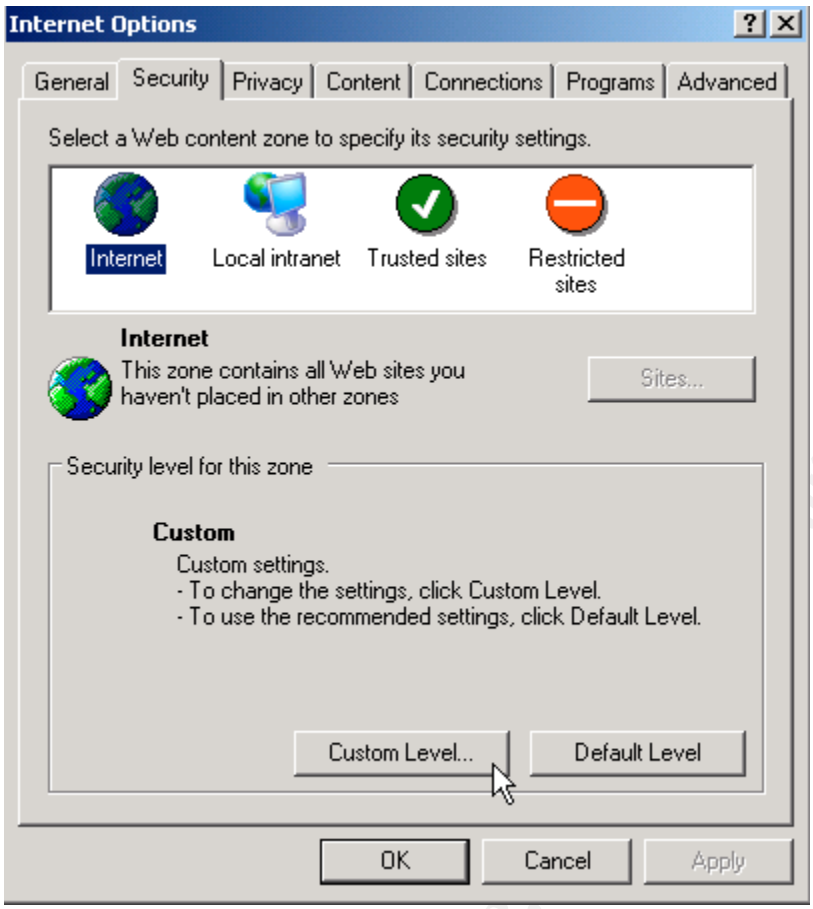
Settings for these zones indicate how content such as ActiveX, Java, and other dynamic content is interpreted. ActiveX, especially, has been demonstrated to be inherently flawed as it places the burden of security on the end user giving them the ability to accept or deny the downloading of scripts to their computer.

> ActiveX puts the responsibility of maintaining the safety of their machines in the hands of the users…I believe that most users are incapable of making informed decisions regarding complicated security risks, and ActiveX is not worth the risk (at least not yet).[22]

It is for this reason that you should seriously consider not even allowing ActiveX past your firewall.

Java settings are the level of security with which the Microsoft Virtual Machine allows Java to run within the browser. While the Java security model is safer than that of ActiveX, it is by no means foolproof. Make sure to keep Java at the highest possible level of security so that rogue java applets are not allowed to run unimpeded on systems.

A miscellaneous section controls whether users can access data sources across domains, submit nonencrypted form data, launch applications and files from IFRAME elements, install desktop items, drag and drop files, copy and paste files, and access software channel features. Each of these settings is important and needs to be understood before customizing your security settings. Within IE, you can access the security settings by going to the Internet Options menu, located under the Tools menu bar at the top of Internet Explorer (see following image).



You have two options within each zone. You can either choose each setting individually through the Custom Level button or use Microsoft's default rating system of Low, Medium-Low, Medium, or High. By using the custom option you have greater flexibility over IE's security. Start with the Microsoft default setting, then click on the custom button to create your settings from the Microsoft base. Microsoft has more information on the details of each setting on its website.[23] My recommendation is to use the custom option on both the Internet Zone and Local Intranet Zone as they are the two zones firms most frequently use. The Internet Zone is the zone for most sites that lie outside of your internal network. The Local Intranet Zone is for most sites on your Intranet. Following is a brief overview of each zone including the default level of security description.

Internet Zone

The settings you choose for the Internet zone define the rules for how IE interacts with sites that lie outside of your internal network. Sites such as Google or Yahoo would typically be in the Internet Zone.

Microsoft Default Security Level: Medium

- Safe browsing and still functional

- Prompts before downloading potentially unsafe content
- Unsigned ActiveX controls will not be downloaded
- Appropriate for most Internet sites

Recommended Customizations:
In the Internet Zone you should disable most of the ActiveX Controls. My recommendation inside of a large firm is to go with the medium settings (with the exception of the Download Signed ActiveX Controls option which, by default, is set to Prompt). The Prompt feature places a lot of trust in the user and inside of a larger firm it can be hard to justify that kind of faith. You should disable that feature for the Internet Zone. The Microsoft VM should be set to high for all java and the default medium settings should be used for the Miscellaneous section.

Local Intranet Zone

The Intranet zone defines the rules for sites within your internal network. Your company's internal website or web applications hosted within your company are typically found behind your company's firewall and are within the Intranet Zone.

Microsoft Default Security Level: Medium-Low

- Same as medium without prompts
- Most content will be run without prompts
- Unsigned ActiveX controls will not be downloaded
- Appropriate for sites on your local network (Intranet)

Recommended Customizations:
In the Intranet Zone you should build off of the default medium settings. Then, change Download Signed ActiveX Controls to Enable, instead of Prompt, as the Prompt feature serves as nothing more than an annoyance on the way to the user accepting the ActiveX control. This should only be enabled if you are behind a firewall. Download settings should both be set to Enable. Any file that is available to download within your intranet should be signed and safe for all users within your firm. Once again, keep the Microsoft VM at high security. Finally, in the Miscellaneous section, I recommend setting Access Data Sources Across Domains to Enable to allow for some of the greater interoperability between IE and new office products.

With the Local Intranet Zone, make sure that you allow more access than with the Internet Zone. By doing this you can give internal developers more power to create web applications that take advantage of active content.

Trusted Sites Zone

Specific sites must be added to the trusted sites zone for the security settings within that zone to be applied. Place only the sites you know and trust completely in this zone.

Microsoft Default Security Level: Low

- Minimal safeguards and warning prompts are provided
- Most content is downloaded and run without prompts
- All active content can run
- Appropriate for sites that you absolutely trust

Recommended Customizations:
None. In this zone you should use Microsoft's default security setting of Low which enables most of the Active Scripting.

Restricted Sites Zone

Microsoft Default Security Level: High

- The safest way to browse, but also the least functional
- Less secure features are disabled
- Appropriate for sites that might have harmful content

Recommended Customizations:

None. This zone is set to Microsoft's default security setting of "High" which disables all of the Active Scripting. Place Internet Sites in here that you know could cause trouble.

To make sure that these settings are applied to all of your users, I recommend that you use the Internet Explorer Administration Kit (IEAK) which allows you to develop a customized build of IE that you can then redistribute throughout your firm.[24] By using this tool you can make sure that everyone has received your customized settings.

In smaller firms, it is much easier to have a tight reign on what sites users are able to visit. In these cases it is best to have some formal process in place to have individual sites added to the Trusted Sites security zone. In medium to larger sized firms it is harder to know what types of sites your users may be visiting, making it difficult to make use of that zone.

Remember, browser security is not a "one-size fits all" proposition. It differs depending on the size of your network and the level of computer savvy your users possess. In some cases, the way to keep yourself protected from a potential problem is not in line with keeping your business productive. There is always a tradeoff between usability and security. IE is no exception to that rule.

How to Keep IE Current With Recent Security Patches

Now that you have established the initial security settings, it is time to make sure that IE is updated with the most current patches available. There are three steps to this process:

- Detecting Needed Patches
- Testing Patches
- Deploying patches

Detecting Needed Patches

Before you update the machines on your network, you need to find out which patches are missing. On the Microsoft site there are two options. You can visit the Windows Update site to keep your machine updated.[25] This will give you the recent patches for IE. Another option is the Microsoft Baseline Security Analyzer which gives you information on what patches are missing on your machine.[26] It will detect all missing patches, including those patches needed for IE. Other automated tools for detecting and deploying patches can be purchased to make your life a little easier. Products such as Ecora Corp.'s Ecora Patch Manager 2.0, PatchLink Corp.'s PatchLink Update 4.0, St. Bernard Software Inc.'s UpdateExpert 6.0, and Shavlik Technologies LLC's HFNetChkPro are leading contenders in the patch management field.[27] These products are not IE specific, but they include IE patches in their repertoire. Each product is able to run detailed inventory scans on your network to let you know which machines are in need of patches.[28] The advantage to this is that you can easily apply the latest patches to machines that are unpatched with the touch of a button from your command console. The drawback is that to apply these patches to a machine requires that you buy a license for that machine. The cost of these products may make it difficult to justify to management. While all of the tools you can purchase allow you greater ease with which to detect and deploy your patches, they do not eliminate the need for testing as they cannot tell you which patches will work properly in your environment.

You should also sign up for the Microsoft Product Security Notification mailing list so that you get bulletins regarding these patches as soon as they come out.[29] Keep on the lookout for potential holes and fixes by staying updated on sites such as SecurityFocus.com and on mailing lists such as Bugtraq. These provide critical information about IE holes and can give you the inside track on upcoming patches. Once the patch is released and you have downloaded it, it is time to start testing.

Testing and Deploying Patches

The best policy for testing the new patches is to establish a repeatable methodology which can be used each time a new patch is released. Before you do any testing, read through the security bulletin and be certain that the patch is

necessary. You should establish a team of individuals who are intimate with the network and can correctly assess the need for the patch. Make sure you get all of the web application developers and testers involved as early as possible in the process. If you have the means to create a testing environment, such as a lab, I recommend this as the way to go. If not, you have to make sure that each one of your application developers has access to the newest patches so that they can test them against their applications.

First, within your lab, make sure you have all of the different builds that you have within your firm. Whether you have a different build for laptops or lines of business, it is imperative that all configurations are tested. Next, create some sample testing scripts for your developers or testers to follow. Test scripts are step-by-step instructions that you want your developers to test. This helps speed up the testing process and gives developers an idea of what to look for while testing their applications. In your scripts, make sure you test all of the vital functions of IE such as printing, accessing an internal website, accessing an external website, and running Java and ActiveX. After all the computers in your lab are set up, schedule times and dates with your developers so that they test them within a certain time frame. Make the deadline is reasonable, but do not give them any leeway in terms of delaying testing. You need to impress upon the firm that this testing is vastly important to the overall security of the network. If you need to test more than one patch, make sure that your testers are installing them in the same order that they will be deployed. Make sure they test their application individually for each patch so that if something does fail you will know which patch has caused the problem.

You also need to have a way of tracking the different web applications that are used in your firm. Know who is responsible for each of those applications and make them understand the importance of security patching. Let all know that they are responsible for testing the patches in a controlled environment against their applications. This will ferret out any potential issues before you release the patch into your environment.

Following is a general outline of steps I created that you should follow to ensure that security patch deployment is done efficiently.

Pre Testing Phase

1. Patch Analysis—Read through the patch manifest. Determine need of patch and timeframe for deployment. Try to predict impact of patch.
2. Communicate Patch—Post a summary of the patch for developers/testers to read.
3. Lab Setup—If feasible, establish a lab for the certification testing of various web applications using all available configurations for your firm.
4. Deployment Strategy—Develop a strategy for the patch deployment. Determine who should get the patches and in what order.

Testing Phase

1. IE Manager Testing—Perform testing of IE before and after installing the patch.
2. Integration Testing—Use established team to test core applications after security patch installation.
3. Test Distribution—Test the deployment of the patches by deploying them to developers/testers.
4. Developer/Tester Testing—Provide a time limit for testing, along with testing guidelines.
5. Track Progress—Track the progress of the testing through e-mails, etc. and post those in a database where everyone can access that information.
6. Deadline Communication—Reiterate the importance of finishing the testing in a timely manner.

Deployment Phase

1. Determine break fixes—Troubleshoot any errors found in testing and restructure deployment date accordingly.
2. Support Staff Communication—Communication to front-line support to let them know of impending deployment and possible issues they may run into.
3. Firmwide Communication—Communicate to firm up the date(s) for deployment of the patches.
4. Deployment—Deploy the patches in the order decided upon during the pre-testing phase.
5. Follow-Up—Post success/failure results of the deployment for developers/testers and support staff.

Once you have initiated the testing process with your developers, you need to have the ability to track their progress. Establish a central repository where you can keep all feedback you receive during the testing. Make this database accessible to all your testers and developers so that they can submit their testing results.

Finally, when deploying the patches make sure that your general user population and support staff are fully aware of the impending release. If you are deploying more than one patch it might be a good idea to leverage some deployment or packaging software if you have it. A good free tool to use is Microsoft's Qchain, which allows you to bundle more than one patch together without having to reboot between the installation of each patch.[30]

> Another headache with manual patch deployment is that a reboot is needed after installation of each patch. Microsoft's QChain eliminates this problem. QChain is a command-line utility that can link multiple hot fixes

together in a single reboot.[31]

By following a testing model such as the one above, you can effectively conduct integration testing of all security patches before releasing them to your general user population.

Conclusion

Internet Explorer is one of the most well known and used applications in the world today. Firms large and small tend to overlook the browser as a potential doorway to sensitive information. Once they put up a firewall they believe that their security work is finished. As I have shown you over the course of this paper, the browser is not inherently secure. Through exploits such as Cross Site Scripting, a malicious attacker can use your unpatched system against you and obtain confidential data by stealing cookies or can even run programs through scripts on your local machine. The best way to prevent these attacks is to maintain a high level of security through customized security settings within IE and to keep your systems regularly patched. The best way to effectively test and distribute these patches is to establish a process that you can use over and over. IE is an integral part of overall network security and should not be overlooked.

List of References

1. "OneStat: Internet Explorer 6 still popular." 16 December 2002. URL: http://www.nua.ie/surveys/index.cgi?f=VS&art_id=905358670&rel=true (31 July 2003).

2. Schnoll, Scott. "The History of Internet Explorer." 1998-2001. URL: http://www.nwnetworks.com/iehistory.htm (31 July 2003).

3. Naraine, Ryan. "When Patches Aren't Applied." CIO Update. 31 March 2003. URL: http://www.cioupdate.com/reports/article.php/2172051 (31 July 2003).

4. Ibid.

5. Ibid.

6. Ibid.

7. Ibid.

8. Seltzer, Larry. "Patch Management: The Enterprise Advantage." Security Supersite. 4 June 2003. URL: http://security.ziffdavis.com/article2/0,3973,1117443,00.asp (31 July 2003).

9. Farber, Dan. "Cybersecurity Report Card – Serious Improvements Needed." ZDNet. 2 June 2003. URL: http://techupdate.zdnet.com/techupdate/stories/main/0,14179,2913868,00.html\ (31 July 2003).

10. Lyman, Jay. "Experts Rip New Microsoft Browser Patch." NewsFactor Network. 16 May 2002. URL: http://www.newsfactor.com/perl/story/17798.html (31 July 2003)

11. "The Twenty Most Critical Internet Security Vulnerabilities (Updated) ~ The Experts' Consensus:W8 Internet Explorer." SANS Institute. Version 3.23. 29 May 2003. URL: http://www.sans.org/top20/#W8 (31 July 2003).

12. "The Cross Site Scripting FAQ." Cgisecurity.com. Article #2. 18 July 2002. URL: http://www.cgisecurity.com/articles/xss-faq.shtml (31 July 2003).

13. Ibid.

14. "Microsoft Security Bulletin MS03-004" Microsoft.com. 19 February 2003. URL:

http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/bulletin/MS03-004.asp (31 July 2003).

15. "showHelp("file:") disables security in IE." SecurityFocus Home Mailing List: BugTraq. 6 February 2003. URL: http://www.securityfocus.com/archive/1/310684 (31 July 2003).

16. Ibid.

17. Sandblad, Andreas. 30 February 2003. URL: http://sandblad.com/security/ (3 August 2003)

18. "showHelp("file:") disables security in IE." SecurityFocus Home Mailing List: BugTraq. 6 February 2003. URL: http://www.securityfocus.com/archive/1/310684 (31 July 2003).

19. Ibid.

20. Ibid.

21. "HTML Help Update to Limit Functionality When It Is Invoked with the window.showHelp() Method." Microsoft Knowledge Base Article 811630. 26 June 2003. URL: http://support.microsoft.com/?kbid=811630 (31 July 2003).

22. "What is ActiveX?" DigiCrime.com. URL: http://www.digicrime.com/activex/ (31 July 2003).

23. "Setting Up Security Zones." Microsoft.com. 7 September 2001. URL: http://microsoft.com/windows/ie/using/howto/security/setup.asp (31 July 2003).

24. "Internet Explorer Administration Kit Homepage." Microsoft.com. 28 July 2003. URL: http://www.microsoft.com/windows/ieak/ (3 August 2003).

25. "Microsoft Windows Update." Microsoft.com. URL: http://www.microsoft.com/windowsupdate (3 August 2003).

26. "Microsoft Baseline Security Analyzer." Microsoft Technet. 4 June 2003. URL:http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/tools/Tools/MBSAhome.asp (3 August 2003).

27. Sturdevant, Cameron. "Solutions Run Gamut." eWeek. 2 June 2003. URL: http://www.eweek.com/article2/0,3959,1115197,00.asp (31 July 2003).

28. Ibid.

29. "Product Security Notification." Microsoft Technet. October 2002. URL: http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/bulletin/notify.asp (1 August 2003).

30. Kull, John. "Develop a strategy for dealing with security patches." CNET Asia. 7 March 2003. URL: http://asia.cnet.com/itmanager/netadmin/0,39006400,39115800-2,00.htm (1 August 2003).

31. Ibid.

Last Updated: January 22nd, 2018

Upcoming SANS Training

Click Here for a full list of all Upcoming SANS Events by Location

SANS Las Vegas 2018	Las Vegas, NVUS	Jan 28, 2018 - Feb 02, 2018	Live Event
Cyber Threat Intelligence Summit & Training 2018	Bethesda, MDUS	Jan 29, 2018 - Feb 05, 2018	Live Event
SANS Miami 2018	Miami, FLUS	Jan 29, 2018 - Feb 03, 2018	Live Event
SANS London February 2018	London, GB	Feb 05, 2018 - Feb 10, 2018	Live Event
SANS Scottsdale 2018	Scottsdale, AZUS	Feb 05, 2018 - Feb 10, 2018	Live Event
SANS SEC455: SIEM Design Beta One 2018	Arlington, VAUS	Feb 12, 2018 - Feb 13, 2018	Live Event
SANS Southern California- Anaheim 2018	Anaheim, CAUS	Feb 12, 2018 - Feb 17, 2018	Live Event
SANS Secure India 2018	Bangalore, IN	Feb 12, 2018 - Feb 17, 2018	Live Event
SANS Brussels February 2018	Brussels, BE	Feb 19, 2018 - Feb 24, 2018	Live Event
SANS Secure Japan 2018	Tokyo, JP	Feb 19, 2018 - Mar 03, 2018	Live Event
Cloud Security Summit & Training 2018	San Diego, CAUS	Feb 19, 2018 - Feb 26, 2018	Live Event
SANS Dallas 2018	Dallas, TXUS	Feb 19, 2018 - Feb 24, 2018	Live Event
SANS New York City Winter 2018	New York, NYUS	Feb 26, 2018 - Mar 03, 2018	Live Event
CyberThreat Summit 2018	London, GB	Feb 27, 2018 - Feb 28, 2018	Live Event
SANS London March 2018	London, GB	Mar 05, 2018 - Mar 10, 2018	Live Event
SANS Secure Singapore 2018	Singapore, SG	Mar 12, 2018 - Mar 24, 2018	Live Event
SANS Secure Osaka 2018	Osaka, JP	Mar 12, 2018 - Mar 17, 2018	Live Event
SANS Paris March 2018	Paris, FR	Mar 12, 2018 - Mar 17, 2018	Live Event
SANS San Francisco Spring 2018	San Francisco, CAUS	Mar 12, 2018 - Mar 17, 2018	Live Event
SANS Northern VA Spring - Tysons 2018	McLean, VAUS	Mar 17, 2018 - Mar 24, 2018	Live Event
SANS Secure Canberra 2018	Canberra, AU	Mar 19, 2018 - Mar 24, 2018	Live Event
SANS Munich March 2018	Munich, DE	Mar 19, 2018 - Mar 24, 2018	Live Event
SANS Pen Test Austin 2018	Austin, TXUS	Mar 19, 2018 - Mar 24, 2018	Live Event
ICS Security Summit & Training 2018	Orlando, FLUS	Mar 19, 2018 - Mar 26, 2018	Live Event
SANS Boston Spring 2018	Boston, MAUS	Mar 25, 2018 - Mar 30, 2018	Live Event
SANS 2018	Orlando, FLUS	Apr 03, 2018 - Apr 10, 2018	Live Event
SANS Abu Dhabi 2018	Abu Dhabi, AE	Apr 07, 2018 - Apr 12, 2018	Live Event
Pre-RSA® Conference Training	San Francisco, CAUS	Apr 11, 2018 - Apr 16, 2018	Live Event
SANS London April 2018	London, GB	Apr 16, 2018 - Apr 21, 2018	Live Event
SANS Zurich 2018	Zurich, CH	Apr 16, 2018 - Apr 21, 2018	Live Event
SANS Baltimore Spring 2018	Baltimore, MDUS	Apr 21, 2018 - Apr 28, 2018	Live Event
SANS Dubai 2018	OnlineAE	Jan 27, 2018 - Feb 01, 2018	Live Event
SANS OnDemand	Books & MP3s OnlyUS	Anytime	Self Paced