ERNW Newsletter 31 / June 2010

Dear Partners, dear Colleagues,

Welcome to the 31th edition of the ERNW Newsletter with the title:

Secure Configuration of Microsoft Internet Explorer, Version 8

Version 1.0, 11th of June 2010

By:
Enno Rey (erey@ernw.de)
Christopher Werny (cwerny@ernw.de)
Oliver Röschke (oroeschke@ernw.de)

Friedwart Kuhn
11.6.2010 17:12:37

Abstract

This newsletter evaluates configuration options, reflecting security and possible usability impact, incorporating typical large scale enterprise usage of browser based content. The evaluation was done for a globally operating enterprise.

Note

We provide some supporting files, namely:

- ❑ some regfiles containing the settings we recommended for the given environment. Use at your own risk and only if you fully understand the impact!
- ❑ an XLS with the recommended settings, based on the Microsoft XLS doc on GPO settings.
- ❑ miscellaneous stuff, e.g. an XLS with some documentation on potentially needed CLSIDs.

A ZIP file with these can be found at
http://www.ernw.de/download/ernw_nl_31_ie8config_supporting_docs.zip
Disclaimer: the recommended settings are deployed in a sufficiently large global network. So they "work in production". Still, your mileage might vary. Please use the stuff cautiously. Any use is at your own risk.

1 INTRODUCTION

Some very large enterprise plans to roll-out *MS Internet Explorer 8* (IE8). Current browser in use is *MS Internet Explorer 6* (IE6). Recent trends show that attackers focus on browser and browser extension vulnerabilities to exploit, as impact as well as number of victims tends to be very high.

Due to the number of security incidents based on browser exploits a secure configuration mitigating common vulnerabilities, threats and subsequent risks is vital for efficient malware protection.

This document describes the main threats, associated risks and derived configuration directives to address the threats while allowing business functionality at the same time.

1.1 Project Goals

To undertake and document an evaluation of configuration options, reflecting security and possible usability impact, incorporating typical large scale enterprise usage of browser based content. The project was lead by a BS 7799 Lead Auditor to ensure compliance of the methodology and recommended measures with the security standard BS 7799/ISO 27000.

1.2 Technical goals

- ❑ Identification of IE8 default configuration settings that are security relevant
- ❑ Analysis and evaluation of associated risks
- ❑ Documentation of secure browser configuration considering certain usage scenarios

The proposed configuration should reflect the following security goals:

- ❑ Integrity and confidentiality of processed data
- ❑ Efficiency and appropriateness, e.g. lowest impact on business functionality
- ❑ Compliance with legal requirements, relevant privacy protection regulations as well as company policies

2 REFERENCES & REQUIREMENTS

2.1 Standards & Best Practice Documents

[CERT_SECURING]

Title: Securing your Web Browser

Source: US Computer Emergency Response Team

Document Scope and Purpose: This document provides introduction to Browser based vulnerabilities and displays practical implementation within main stream browsers.

Source: http://www.cert.org/tech_tips/securing_browser/

[BROWSER_SEC]

Title: Browser Security Handbook

Source: http://code.google.com/p/browsersec/

Document Scope and Purpose: Provides web application developers, browser engineers, and information security researchers references to key security properties of contemporary web browsers.

2.2 Vendor Documents

[MS_IE8_GPSettings]

Group Policy Settings Reference for Windows Internet Explorer 8

Link: http://www.microsoft.com/downloads/details.aspx?FamilyID=ab4655f2-0a3c-42eb-974d-24b2790bf592&displaylang=en

[MS_IE_ADVSettings]

Internet Explorer security zones registry entries for advanced users

Link: http://support.microsoft.com/kb/182569

[MS_IE8_SEC_GUIDE]

Internet Explorer 8 Desktop Security Guide

http://www.microsoft.com/downloads/details.aspx?displaylang=en&FamilyID=44405777-51b4-4376-9cef-f0341b13fcde

[AD_APP_SEC_ACROBAT]

Application Security for the Acrobat Family

http://learn.adobe.com/wiki/download/attachments/64389123/AcrobatApplicationSecurity.pdf?version=1

2.3 Miscellaneous documents referenced here

[GHOST_IN_BROWSER]
The Ghost In The Browser: Analysis of Web-based Malware, Niels Provos, Dean McNamee, Panayiotis Mavrommatis, Ke Wang, Nagendra Modadugu, First Workshop on Hot Topics in Understanding Botnets (HotBots '07), 2007.
Link: http://www.usenix.org/event/hotbots07/tech/full_papers/provos/provos.pdf

3 Sample Websites

This section lists the sample websites that were (all) used for functionality testing, with different parameter sets applied.

3.1 Sample websites proposed by the evaluators

These are the following:

- ❑ www.youtube.com without login
- ❑ www.bahn.de without login
- ❑ www.opodo.de without login
- ❑ www.lufthansa.de with login
- ❑ www.xing.de with login
- ❑ www.volksbank-kurpfalz.de with login

3.2 Sample websites provided by company

These are the following:

- ❑ http://www.sueddeutsche.de/
- ❑ http://www.munich-airport.de/de/consumer/
- ❑ http://dict.leo.org/
- ❑ http://www.spiegel.de/
- ❑ http://www.woerterbuch.info
- ❑ http://web3.hrs.com/web3/
- ❑ http://www.berlin-airport.de

- ❑ All of those were tested *without* logins.

4 OVERVIEW OF BROWSER BASED SECURITY PROBLEMS

This section describes the most important attack vectors against browsers. The configuration directives provided in later chapters are meant to address mainly these threats.

4.1 Malware

Malware is one of the most critical threats for decades, nevertheless it has changed a lot in the last years and is focusing more and more at the most important programs, that are used on the client side like the internet browser. The basic functionality of the browser gets extended today by additional 3rd party software to integrate parsing of the all needed file formats into the browser, implement additional function or running fat client software in the context of the browser. Common files are

- ❑ PDF files
- ❑ Multimedia files (audio, video, pictures)
- ❑ Active X Controls
- ❑ Java Applets
- ❑ Browser Helper Objects

As they invoke the 3rd party software, security vulnerabilities in that software or embedded code in the files can also result in malware infection of the client using the browser as an attack vector.

4.2 XSS

Typically non-existing or incomplete input and output validation in web applications is the main cause of so called cross-site scripting (XSS) problems. The script code is transmitted or "reflected" to the browser of the client. The code is executed in the browser context with the privileges of the logged-in user. Due to the enriched function set of modern script languages even backdoors can be developed in programming languages as JavaScript and clients can be infected just by using a vulnerable web application. Other OS related vulnerabilities of the infected client may also result in privilege escalation attacks and a subsequent total compromise of the client.

4.3 Web 2.0

Web 2.0 introduced more client side functionality to improve interaction with the web application located on the server; examples are better integration into the user desktop and asynchronous communication with the server to overcome the limitations of the "old-style" web applications. AJAX, Adobe Flash und Microsoft Silverlight are technologies that are used to develop the so called *Rich Internet Applications* (RIA) and extend the possibilities of software development to the features of dedicated client applications like word processors and typical business applications (e.g. customer relationship, employee management). Of course current browsers must support these functionalities and might run these web 2.0 applications from untrusted sources in the internet, which might result in installation of backdoors, data theft and even identity theft. Basic web 2.0 technology is already provided by the features that JavaScript offers.

4.4 Information Disclosure due to Caching

For an increase of performance browsers are caching downloaded content to access it faster when the same information is requested again. Also sensitive content is cached on the client side and can be revealed by accessing the user's browser cache.

4.5 Cookies

Depending on their use cookies can be subject of two main kinds of attack. When used for authentication or session state, they can be stolen for session hijacking and authentication attacks, so the browser must protect them properly and ensure that they can't be read by untrusted sources. The other type of attack is user tracking. Tracking web sites set cookies using advertisements embedded in web sites to track what the user is doing and what kind of web sites she or he is visiting. The gathered information is then used for profiling the user behaviour and can also be used for preparing targeted attacks.

4.6 iFrames

An iFrame (this stands for "inline frame") is just a way of loading one web page inside another, usually from a different server, looking like one application at the same time. This can be useful for building online applications, but it can be also abused by malware writers that make the included page just appear like a one pixel square – meaning it's not visible to the user – and obfuscate JavaScript that will run automatically from that included page and results in the same kind of attacks as cross-site scripting.

4.7 Browser Based Buffer Overflows

Buffer overflow attacks are getting less important these days due to the security development lifecycle that vendors are implementing to make their software (in particular their operating systems) more secure. So attackers are moving to applications that are running on top of the operating system like browsers and their plug-ins and add-ons. Due to the rich feature sets browsers are the target of choice today and many discovered buffer overflow vulnerabilities are related to the browser. Compromising the system, targeting buffer overflows in browsers and their extensions, only compromises the client in the context of the working user, but privilege escalation attacks might lead to a total compromise of the system. The problem gets worse as long as the vendors of the extensions don't use the additional security functions, that e.g. Microsoft provides, like DEP, ASLR and SafeSEH.

4.8 Drive-by-Downloads

The term *drive-by-download* describes the unintended download and installation of malicious software on a computer. It usually means that malware is installed just by visiting a website. This method of installing malware on computers utilizes technologies in the browser that are used to display active content, for example JavaScript or Flash based content. These technologies might not be allowed to directly access resources on the computer per se, they are running in a sandbox in the browser, thus isolation the computer from active web content. The wide distribution of drive-by-downloads is possible by the fact that all browsers have a more or less long history of security vulnerabilities that allows code to break out the sandbox and use resources aside the browser sandbox.

After escaping from the browser sandbox the code downloads additional modules or new malware and installs it on the computer. From an attackers point of view the following essential steps are necessary to accomplish a successful drive-by-download attack:

❑ Add a (malicious) content to a website, therefore mostly public websites are attacked, these websites are modified in a way to include active content from a webserver that is controlled by the attacker and hosts the active content. The most common way to include content in the website is to embed it in an iFrame.

- Get users to visit the site, this can be done for example via email if it is necessary at all, depending on the degree of popularity it is optional to attract additional users.
- Deliver the malware for the computers that are successfully exploited and requesting the malware

[GHOST_IN_BROWSER, section 5.1] describes a typical drive-by-download that is split into various stages:

- The exploit is delivered to a user’s browser via an iFrame on a compromised web page.
- The iFrame contains Javascript to instantiate an ActiveX object that is not normally safe for scripting.
- The Javascript makes an XMLHTTP request to retrieve an executable.
- Adodb.stream is used to write the executable to disk.
- A Shell.Application is used to launch the newly written executable.

Even though some of the mentioned technologies (e.g. the ADO.DB method) are banned from browsers (IE) in the interim this gives an idea of the way drive-by-downloads work.

5 MAIN STEPS OF SECURING IE8

The following steps were identified as useful approaches for mitigating the risks arising from the threats described above:

- ❑ Use OS's memory protection mechanisms (at least for the browser process).
- ❑ Wherever possible, use 64-bit browser
- ❑ Perform hardening of core browser environment
- ❑ Control add-on behaviour
- ❑ Use additional browser security capabilities
- ❑ Use network level security capabilities wherever possible
- ❑ Prevent users from modifying options

These are described in more detail in the following chapter.

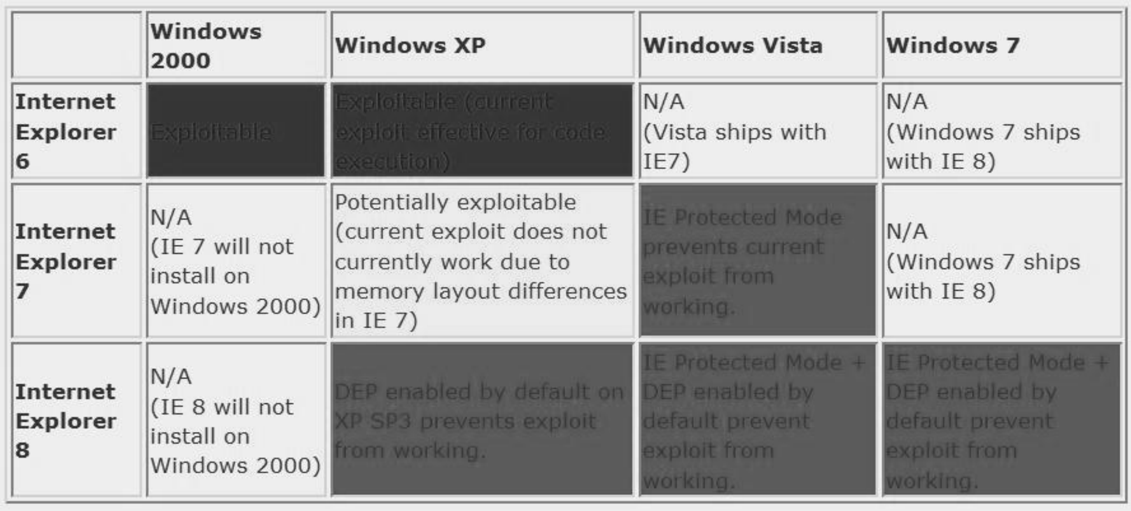
5.1 Use OS's memory protection mechanisms, namely DEP

Data Execution Prevention (DEP) is a security feature included in modern Microsoft Windows operating systems that is intended to prevent an application or service from executing code from a non-executable memory region. This helps prevent certain exploits that store code via a buffer overflow, for example. DEP runs in two modes: hardware-enforced DEP for CPUs that can mark memory pages as non-executable, and software-enforced DEP with a limited prevention for CPUs that do not have hardware support.

DEP was introduced in Windows XP Service Pack 2 and is included in Windows Server 2003 Service Pack 1 and later, Windows Vista, and Windows Server 2008, and all newer versions of Windows.

While DEP does not provide a "silver bullet" to prevent all memory corruption attacks (e.g. buffer overflows) and the authors are well aware of exploits circumventing MS Windows memory protection mechanisms to some degree it can safely be assumed that the effort to code an exploit working on a DEP enabled platform is much higher than on a platform not using DEP. Following the insight that exploit code authors want to target the largest possible user base with a given amount of coding effort, as of early 2010 it can still be expected that exploit code primarily is developed for platforms without DEP.

For example some MS blog entry[1] discussing the IE security problems published in January 2010 (and forcing Microsoft to release an "urgent patch") lists



	Windows 2000	Windows XP	Windows Vista	Windows 7
Internet Explorer 6	Exploitable	Exploitable (current exploit effective for code execution)	N/A (Vista ships with IE7)	N/A (Windows 7 ships with IE 8)
Internet Explorer 7	N/A (IE 7 will not install on Windows 2000)	Potentially exploitable (current exploit does not currently work due to memory layout differences in IE 7)	IE Protected Mode prevents current exploit from working.	N/A (Windows 7 ships with IE 8)
Internet Explorer 8	N/A (IE 8 will not install on Windows 2000)	DEP enabled by default on XP SP3 prevents exploit from working.	IE Protected Mode + DEP enabled by default prevent exploit from working.	IE Protected Mode + DEP enabled by default prevent exploit from working.

More information about DEP in the context of MS IE8 can be found here:

http://blogs.msdn.com/ie/archive/2008/04/08/ie8-security-part-I_3A00_-dep-nx-memory-protection.aspx

5.1.1 Recommendations

It is STRONGLY RECOMMENDED to leave DEP enabled for the Internet Explorer process (it is activated *by default* for IE8 running on XP SP3). From the authors experience in different corporate environments no subsequent functionality problems arise; problems presumably related to DEP should be tracked for different origins.

On 64-bit platforms DEP can't be disabled (see link above).

[1] *http://blogs.technet.com/srd/archive/2010/01/15/assessing-risk-of-ie-0day-vulnerability.aspx*

5.2 64-bit vs. 32-bit Internet Explorer

It should be noted that some add-ons (and subsequently the security problems potentially induced by them) only exist as 32-bit variants (e.g. Adobe Flash as of time of writing). Furthermore – following the insight that exploit code authors want to target the largest possible user base with a given amount of coding effort – as of early 2010 it can still be expected that exploit code primarily is developed for the 32-bit variant.

Last but not least DEP can't be disabled on 64-bit platforms.

5.2.1 Recommendations

It is recommended to instruct users to primarily use the 64-bit variant (and temporarily switching to the 32-bit version for websites needing add-ons not available for the 64-bit browser [e.g. Adobe Flash]).

The authors are well aware of this causing a slight amount of confusion or annoyance within the user community. Still it should be noted that the expected security benefits heavily outweigh the impact on the user experience.

5.3 Hardening of core browser environment

This is described in detail in the next chapter.

5.4 Control add-on behaviour

This section discusses the main types of add-ons, their associated security problems and how to (partially) address them.

5.4.1 Browser Helper Objects

A Browser Helper Object (BHO) is a DLL module designed as a plugin for Microsoft's Internet Explorer web browser to provide added functionality. BHOs were introduced in October 1997 with the release of version 4 of Internet Explorer. Most BHOs are loaded once by each new instance of Internet Explorer. However, in the case of the Windows Explorer, a new instance is launched for each window.

Some modules enable the display of different file formats not ordinarily interpretable by the browser. For example the Adobe Acrobat add-on allows Internet Explorer users to read PDF files within their browser. Other modules add toolbars to Internet Explorer, such as the *Alexa Toolbar* that provides a list of web sites related to what a user is currently browsing, or the *Google Toolbar* that adds a toolbar with a Google search box to the browser user interface.

The BHO API exposes hooks that allow the BHO to access the Document Object Model (DOM) of the current page and to control navigation. Because BHOs have unrestricted access to the Internet Explorer event model, some flavors of malware have also been created as BHOs. For example, the *Download.ject* malware installs a BHO that would activate upon detecting a secure HTTP connection to a financial institution, record the user's keystrokes (intending to capture passwords) and transmit the information to a website used by Russian computer criminals.

In response to the problems associated with BHOs and similar extensions to Internet Explorer, Microsoft added an Add-on Manager in Internet Explorer 6 with the release of Service Pack 2 for Windows XP (updating it to IE6 Security Version 1). This utility displays a list of all installed BHOs, browser extensions and ActiveX controls, and would allow the user to enable or disable them independently. Given the requirements stated above, in the company environment it is not planned to give users control over add-ons though.

5.4.2 ActiveX

[CERT_SECURING, section II] gives the following overview:

"ActiveX is a technology used by Microsoft Internet Explorer on Microsoft Windows systems. ActiveX allows applications or parts of applications to be utilized by the web browser. A web page can use ActiveX components that may already reside on a Windows system, or a site may provide the component as a downloadable object. This gives extra functionality to traditional web browsing, but may also introduce more severe vulnerabilities if not properly implemented.

ActiveX has been plagued with various vulnerabilities and implementation issues. One problem with using ActiveX in a web browser is that it greatly increases the attack surface, or 'attackability' of a system. Installing any Windows application introduces the possibility of new ActiveX controls being installed. Vulnerabilities in ActiveX objects may be exploited via Internet Explorer, even if the object was never designed to be used in a web browser (VU#680526). In 2000, the CERT/CC held a workshop to analyze security in ActiveX. The results from that workshop may be viewed at http://www.cert.org/reports/activeX_report.pdf. Many vulnerabilities with respect to ActiveX controls lead to severe impacts. Often an attacker can take control of the computer. You can search the Vulnerability Notes Database for ActiveX vulnerabilities at http://www.kb.cert.org/vuls/byid?searchview&query=activex."

The (rated "critical") advisory MS09-032 (*Vulnerability in Microsoft Video ActiveX Control Could Allow Remote Code Execution*) serves as an excellent example of an ActiveX control presumably unneeded in corporate environments but leading to a potentially business critical attack surface.

Subsequently the use of ActiveX controls should be strictly controlled/restricted.

5.4.3 Controlling add-ons

Here the main parameter is "Deny all add-ons unless specifically allowed in the Add-on List" (see [MS_IE8_GPSettings] for the exact registry key that is *independent* of the *zone*). "Enabling" this parameter allows to control the use of ActiveX controls. If enabled without further listing allowed add-ons by means of their respective CLSIDs essentially disables any use of add-ons. Thus listing CLSIDs allows for a fine-grained control of add-ons as a whole.

It should be noted that the authors regard this parameter as one key element for overall browser (IE) security.

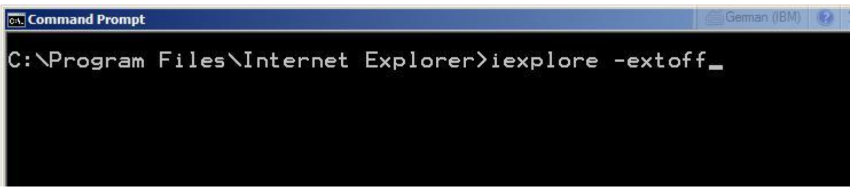
5.4.4 On the role of CLSIDs and important examples

As stated above the explicit listing of "allowed" CLSIDs is the only way of providing their use once the overall usage is limited by enabling the parameter. Therefore if choosing the "restriction approach" (which, again, is recommended here), it is crucial to be able to identify potentially needed add-ons when initially deploying the feature and being able to track new add-ons over time by appropriate (operational) processes.

The CLSID of any given add-on can be identified by right-clicking the respective add-on in the add-on manager. Furthermore in the appendix of this document a regfile containing all CLSIDs present in a default installation of IE8 on the test platform (see appendix for details) can be found.

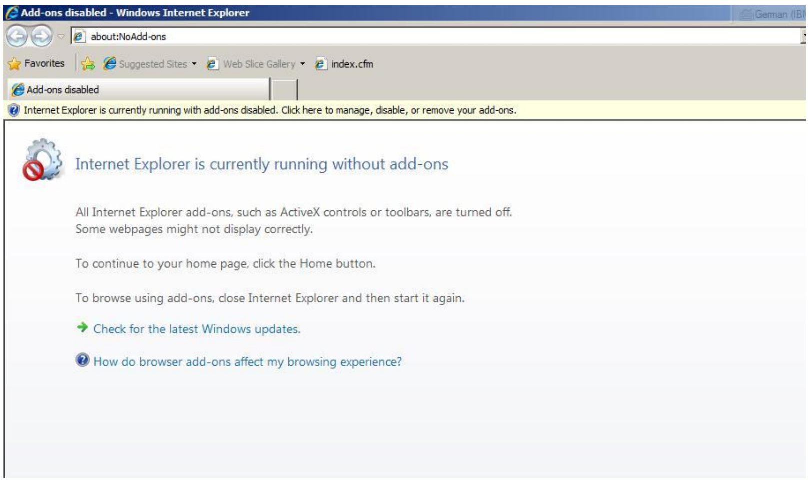
5.4.5 Testing if a website needs add-ons

Besides "traditional functionality testing" by simple observing a website's behaviour the Internet Explorer can be started without loading any add-ons by the following command line switch:



```
C:\Program Files\Internet Explorer>iexplore -extoff_
```

IE then starts like this



Internet Explorer is currently running with add-ons disabled. Click here to manage, disable, or remove your add-ons.

Internet Explorer is currently running without add-ons

All Internet Explorer add-ons, such as ActiveX controls or toolbars, are turned off. Some webpages might not display correctly.

To continue to your home page, click the Home button.

To browse using add-ons, close Internet Explorer and then start it again.

Check for the latest Windows updates.

How do browser add-ons affect my browsing experience?

which in turn can be used to observe the behaviour of certain websites without any add-ons.

5.4.6 DEP for add-ons

As (the vast majority of) add-ons run in the process context of IE, they are covered by IE's memory protection mechanisms as well. Still DEP should be enabled independently for some important add-ons (see below).

5.4.7 Control Approaches

The authors see three main approaches with regard to handling add-ons that are defined in the following.

It should be noted that the control/restriction of add-on usage by means of the parameter described above is *not* dependent on the zone of a website. Unfortunately this means that add-on usage can not be controlled on a zone level (which otherwise would have provided the opportunity to, say, leave a liberal approach of handling add-ons just for "trusted sites".)

5.4.7.1 Strict with few add-ons

Here, add-on usage is generally restricted and only a few add-ons are allowed (by listing their CLSIDs) at all. As of the authors observation a total of about 8–10 add-ons (including for example JAVA and Adobe add-ons) should be sufficient to use nearly all websites (needed in corporate environments) without major loss of functionality. Still the authors are aware that this approach might impose either too many restrictions or too much operational (tracking) effort in very volatile environments (like the company corporate network).

5.4.7.2 Strict with many add-ons

Here, add-on usage is generally restricted and a larger number (usually between 30 and 50) of add-ons/CLSIDs is enabled. From the observations during testing it can be stated that this generally provides a very good user experience even in environments asking for highly dynamic (website) content.

5.4.7.3 Liberal approach

Here, the "Deny all add-ons unless specifically allowed in the Add-on List" parameter is not used at all (thereby either potentially enabling flexible download/use of add-ons or mandating for other operational processes for tracking them [like whitelisting of binaries which is certainly out of scope currently] on corporate desktops). Still, it is possible to restrict the *download* of ActiveX controls by other means (parameters) on a zone level (see below). However this would not prevent the exploitation of ActiveX already deployed (by whatever potentially unrelated installation process) on desktops.

5.4.8 Recommendations

The authors recommended the following for the company environment:

Scenario / template	**Recommended approach**
Computers protected by additional MCP	"Strict with many" recommended, "liberal" if absolutely needed. From authors assessment, a risk acceptance should be filed in the latter case.
Computers without additional MCP	"Strict with few" recommended, "strict with many" if absolutely needed. Do NOT go with "liberal" as high business risk imposed.

5.5 Use enhanced browser security capabilities

Internet Explorer 8 itself provides some enhanced security modules which may be of use for protecting IE usage in corporate environments. The two most important of those are discussed in the following.

5.5.1 XSS Filter

Internet Explorer's Cross-Site Scripting (XSS) filter can help prevent one website from adding script code to another website. The Internet Explorer 8 XSS filter tries to protect from Type 1 (also known as "reflected") XSS attacks. This is done by building signatures from the request by scanning the request and matching it against heuristics. The response is then scanned for these signatures. If a signature of active content is found in the response a possible XSS attack has been detected. In this case the Internet Explorer changes the content of the website to prevent the execution of the active content.

It is possible for the webserver to deactivate the XSS filter, for its domain, on the client (Internet Explorer) by sending a special HTTP header. This is done for compatibility with websites that depend on the reflection user input (mostly these sites are vulnerable to XSS attacks). From a security point of view this is not desirable, but the XSS filter in general still has security value, even with the possibility for the server to deactivate it or to circumvent it by obfuscating the XSS content. For an attacker it is much harder and also a not so common vulnerability to be able to inject HTTP header information than executing a XSS attack. The following links provide more detailed

information regarding the Internet Explorer 8 XSS filter.

http://blogs.technet.com/srd/archive/2008/08/19/ie-8-xss-filter-architecture-implementation.aspx

http://blogs.msdn.com/ie/archive/2008/07/01/ie8-security-part-iv-the-xss-filter.aspx

http://msdn.microsoft.com/en-us/library/dd565647%28VS.85%29.aspx

http://blogs.msdn.com/dross/archive/2008/07/03/ie8-xss-filter-design-philosophy-in-depth.aspx

5.5.1.1 Recommendation

XSS filter should be used on corporate desktops without allowing user opt-out.

5.5.2 SmartScreen

SmartScreen Filter is a feature intended to detect phishing websites and to prevent malware installation. It operates in the background while browsing the web, analyzing webpages and determining if they have any characteristics that might be suspicious. It checks visited sites against a dynamic list of reported phishing sites and malicious software sites.

If SmartScreen Filter is enabled, it first checks the address of the website which is visited against a list of high traffic website addresses stored on the computer that Microsoft believes to be of legitimate use. Addresses that are not on the local list and those of downloaded files are sent to Microsoft and checked against a frequently updated "blacklist" of websites and downloads

5.5.2.1 Recommendation

Due to privacy concerns (data sent to MS allowing some tracking) not to be enabled.

5.6 Use network level security capabilities

Wherever possible additional network level controls like content filtering devices, proxies disposing of anti-malware features and the like should be used. Given the multitude of possible attacks paths in the browser context this should be mandatory in any corporate environment.

If not available, the authors recommend to file a risk acceptance to allow/require a change within a reasonable time frame.

5.7 Prevent users from modifying options

Due to the complexity of browser based technologies and their associated security aspects users should not be allowed to change any settings or take whatever kind of (security related) decision on their own.

6 HARDENING OF CORE BROWSER ENVIRONMENT

This section discusses how the core browser environment of MS IE8 can be configured with tightened security. There's a huge number or parameters that can be set via the Internet Explorer GUI, by Group Policy (GP[O]) or by modifying the windows registry directly.

Currently, to the knowledge of this paper's authors there is no comprehensive documentation of these parameters available and their interaction between/amongst them and the behaviour of websites is not too well understood. However, to some degree this was compensated by extensive testing when this paper was written.

To add even more confusion some of this settings apply to individual zones (see below) while others affect the behaviour of IE regardless of the website accessed (and associated zone). Furthermore the configuration of parameters is done at different locations throughout the Windows registry.

All parameters discussed herein are marked and commented on in the accompanying data sheet (which, in turn, is based on [MS_IE8_GPSettings])

6.1 Internet Explorer Zones

It is assumed the reader disposes of a rough understanding of the zone concept of Internet Explorer. Thus only a short description is given here.

6.1.1 Internet Zone

The kind-of-default zone for all external websites visited, usually identified by a fully qualified domain name.

6.1.2 Trusted Sites

Here sites regarded trustworthy (and subsequently to be less restricted as for their behaviour) can be added. This might lead to operational overhead and to slow startup of Internet Explorer. The authors recommend to keep the number of "trusted sites" small and the "internet zone" configuration approach described below reflects that in the sense that most sites should work in the *internet zone* anyway.

Furthermore, quoting from [BROWSER_SEC_HANDBOOK], it should be noted that:

"The model fails to account for the impact of cross-site scripting flaws in trusted sites. Since users add pages such as Windows Update, their banks, and other legacy sites that may not always work properly in the 'Internet' zone to 'Trusted sites', giving them a lot of unnecessary permissions as a side effect, the impact of cross-site scripting flaws on these pages may result in the attacker suddenly gaining the ability to carry out dangerous actions normally not available to Internet content."

6.1.3 Local Intranet

By default, the Local Intranet zone contains all network connections that were established by using a Universal Naming Convention (UNC) path, and Web sites that bypass the proxy server or have names that do not include periods (for example, http://local), as long as they are not assigned to either the Restricted Sites or Trusted Sites zone.

Given the role and significance of the *Local Intranet* zone and the subsequent impact as for intranet applications or HTTP based administrative ("web") interfaces of e.g. network devices, the authors suggest not touching the parameters of this zone and leaving it in the default state. There is no relevant risk expected from this approach.

An overview of the default settings of some parameters for the most important zones can be found here:

http://blogs.technet.com/sterileey/archive/2008/09/16/internet-explorer-security-levels-compared.aspx

While this article refers to IE 7, from the authors' perspective it can be assumed that the vast majority of these parameters has *not* changed in IE 8.

In the following all parameters configurable for security purposes are grouped according to their control approach and associated security benefit. The exact settings can be found in the accompanying table. All parameters have been extensively tested on the sample websites listed above. When not listed explicitly as a "functionality impacting parameter" (see extra section on this) it can be assumed that the "[security wise] strict setting" of some parameter does not impair the usability of the vast majority of websites.

6.2 Parameters controlling scripting behaviour

These parameters allow to control the general scripting activity. Wherever possible (that means not leading to major usability impact) scripting activity should be minimized as any script operation may lead to security problems.

The following parameters belong to this category:

Allow active content over restricted protocols to access my computer
Allow active scripting
Allow binary and script behaviours
Allow cut, copy or paste operations from the clipboard via script
Allow scripting of Internet Explorer web browser control
Allow Scriptlets
Allow status bar updates via script
Allow video and animation on a webpage that does not use external media player
Java permissions
Scripting of Java applets

6.3 Parameters controlling downloads and/or installation actions

Obviously any download or even ("code") installation activity should be minimized to the largest possible extent. There are a number of parameters suited to control this.

The following parameters belong to this category:

Allow file downloads
Allow font downloads
Allow Install On Demand (Internet Explorer)
Allow installation of desktop items
Allow software to run or install even if the signature is invalid
Disable .NET Framework Setup
XAML browser applications

XPS documents
Check for signatures on downloaded programs

6.4 Parameters controlling ActiveX behaviour

Looking at the numerous attack paths induced by running ActiveX controls, again, following the "minimal machine" principle (that means: only allowing absolutely needed components to run) seems a reasonable approach.

The following parameters belong to this category:

Download signed ActiveX controls
Initialize and script ActiveX controls not marked as safe
Run ActiveX controls and plugins
Script ActiveX controls marked safe for scripting
Download unsigned ActiveX controls
Turn off ActiveX opt-in prompt
Only allow approved domains to use ActiveX controls without prompt
Only use the ActiveX Installer Service for installation of ActiveX Controls
Disable Per-User Installation of ActiveX Controls
Automatic prompting for ActiveX controls

6.5 Parameters controlling add-on behaviour

On a more general level the overall behaviour of add-ons can be controlled (see also discussion on add-ons above).

The following parameters belong to this category:

Allow third-party browser extensions
Deny all add-ons unless specifically allowed in the Add-on List

6.6 Parameters preventing users from modifying settings or similar actions

Wherever possible, user interaction (or even user control over security settings) should be avoided. There are a huge number of parameters somehow contributing to this goal.

The following parameters belong to this category:

Automatic prompting for file downloads
Prevent Bypassing SmartScreen Filter Warnings
Display mixed content
Launching programs and unsafe files
Turn off Managing SmartScreen Filter
Prevent ignoring certificate errors
Use Pop-up Blocker
Use SmartScreen Filter
Security Zones: Do not allow users to add/delete sites

Security Zones: Do not allow users to change policies
Software channel permissions
Do not prompt for client certificate selection when no certificates or only one certificate exists
Do not allow users to enable or disable add-ons
Turn off Managing Pop-up Allow list
Turn off managing Pop-up filter level
Disable change of proxy settings
Prevent setting of the code download path for each machine
Disable Security Page
Disable the (Dial-Up) Connections Page
Disable the Content page
Disable the General page
Disable the Privacy page
Disable the Programs page
Allow active content from CDs to run on user machines
Do not allow resetting Internet Explorer settings
Turn off configuring the update check interval (in days)
Information bar
Disable AutoComplete for forms
Disable AutoComplete for usernames/passwords
Warn if changing between secure and not secure mode
Allow active content to run in files on My Computer
Enable DOM Storage
Turn Off First-Run Opt-In
Prevent the configuration of cipher strength update information URLs
Warn if POST submittal is redirected to a zone that does not permit posts

6.7 Logon Behaviour

Logon behaviour can be changed within different zones, to reflect trustworthiness of logon credentials to certain web sites.

If one enables/modifies this policy setting, one can choose from the following logon options.

- ❑ Anonymous logon

  Disable HTTP authentication and use the guest account only for the Common Internet File System (CIFS) protocol.
- ❑ Prompt for user name and password

  Query users for user IDs and passwords. After a user is queried, these values can be used silently for the remainder of the session.
- ❑ Automatic logon only in Intranet zone

  Query users for user IDs and passwords in other zones. After a user is queried, these values can be used silently for the remainder of the session.
- ❑ Automatic logon with current user name and password

  To attempt logon using Windows NT Challenge Response (also known as NTLM authentication). If Windows NT Challenge Response is supported by the server, the logon uses the user's network user name and password for logon. If Windows NT Challenge Response is not supported by the server, the user is queried to provide the user name and password. If one disables this policy setting, logon is set to Automatic logon only in Intranet zone.

If these policy settings are not configured, logon is set to *Prompt for username and password*.

To change this setting, the following Registry Key must be modified:

HKLM\Software\Policies\Microsoft\Windows\CurrentVersion\Internet Settings\Zones\3 and edit REG_DWORD parameter 1A00, with one of the following values:

- ❑ Value of 00000 = Automatic logon only with current username and password
- ❑ Value of 10000 = Prompt for username and password
- ❑ Value of 20000 = Automatic logon only in the Intranet Zone
- ❑ Value of 30000 = Anonymous logon

This setting affects how data entered into login fields is handled by Internet Explorer. It is recommended (and required by the company) to always ask for username and password within the *Internet Zone* and to avoid saving the credentials of past sessions.

6.8 Privacy related parameters

Some parameters are initially meant to provide enhanced privacy, but can further be used for security purposes.

The following parameters belong to this category:

Turn off InPrivate Browsing
Empty Temporary Internet Files folder when browser is closed
Userdata Persistence

6.9 Miscellaneous

These settings include various parameters that might have security benefit when configured appropriately but do not fit into one of the categories used above.

The following parameters belong to this category:

Do not save encrypted pages to disk
Include local directory path when uploading files to a server
Loose XAML files
Navigate windows and frames across different domains
Allow websites to open windows without address or status bars
Allow websites to prompt for information using scripted windows
Launching applications and files in an IFRAME
Open files based on content, not file extension
Submit non-encrypted form data
Web sites in less privileged Web content zones can navigate into this zone
Turn on Cross-Site Scripting (XSS) Filter
Check for server certificate revocation
Run .NET Framework-reliant components not signed with Authenticode
Run .NET Framework-reliant components signed with Authenticode
Internet Zone Restricted Protocols
Send internationalized domain names
Automatically check for Internet Explorer updates
Play animations in web pages
Play sounds in web pages
Turn off Encryption Support
Turn off Cross Document Messaging
Turn off the XDomainRequest Object
Enable Native XMLHttpRequest Support
Consistent MIME handling
Enable Zone Elevation Protection
MIME sniffing
Object Caching
Security Zones: Use only machine settings
Enable memory protection to help mitigate online attacks
Warn about certificate address mismatch
Check for publisher's certificate revocation
Enable integrated Windows Authentication
Access data sources across domains
Allow drag and drop or copy and paste files
Allow META REFRESH
Allow script-initiated windows without size or position constraints

6.10 Parameters with huge functionality impact

In the course of the evaluation and testing phase a small number of parameters was identified that had particularly huge functionality impact in the sense that activating them caused usability problems with several websites from the sample set.

To provide a better understanding of these and the reasoning behind the settings recommend, they are listed and discussed in the following.

6.10.1 Allow active scripting

After disabling this parameter, most of the tested sites had a major decrease in functionality, since nearly all of them need JavaScript. Even though certainly desirable from a security point of view, disabling this is regarded impossible in a volatile corporate environment.

6.10.2 Run ActiveX controls and plugins

Disabling this option prevents ActiveX controls/plug-ins from running. Again a huge impact on most tested websites could be observed.

6.10.3 Script ActiveX controls marked safe for scripting

This option determines whether an ActiveX control marked safe for scripting can interact with a script. The behaviour resulting from disabling is mostly the same as for the previous parameter.

6.10.4 Submit non-encrypted form data

If this parameter is enabled, form data can only be sent over an encrypted connection like SSL or TLS. So, if some website doesn't provide a transmission with https, no requests with form data can be sent to the webserver. This has huge (negative) impact on many websites used in corporate context (e.g. from the sample websites used here, on berlin-airport.de the search function for flights doesn't work anymore).

6.10.5 Deny all add-ons unless specifically allowed in the Add-on List

After activating this parameter *without* allowing appropriate add-ons at the same time, a huge impact could be observed, such as no more working tools at tested websites. After allowing the appropriate add-ons, the sites are working as intended. See also discussion on this above.

6.10.6 Enable Native XMLHttpRequest Support

This parameter controls one of the main features of AJAX, that is the ability to request data from a server and to load it into the current website, without the need to reload the entire site. Unfortunately disabling this parameter has vast impact on a number of websites (e.g. googlemaps relies on it). From the sample websites, on web3.hrs.com no matching destinations are shown while entering characters and the routing at “Map-based search” doesn't work.

6.10.7 Recommendations for these parameters

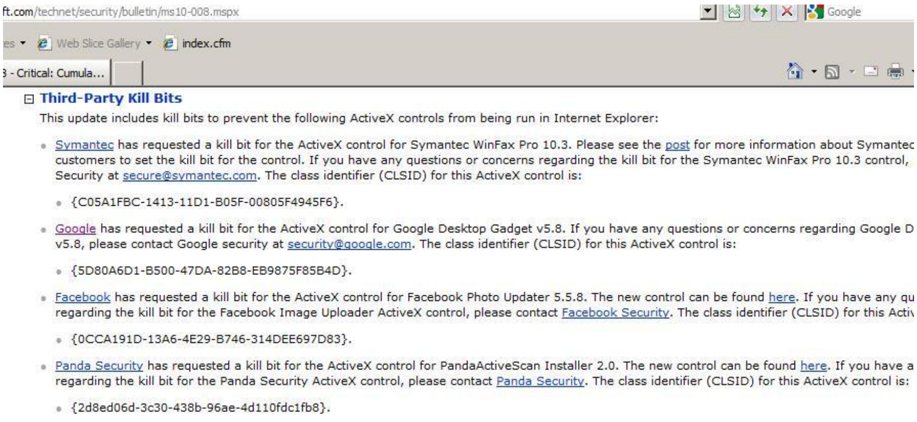
Given the inherent conflict of usability requirements and security requirements, the following recommendations are given as for different deployment scenarios:

Parameter	**Recommended setting for internet zone with MCP**	**Recommended setting for internet zone without MCP**	**Recommended setting for trusted sites**
Allow active scripting	Enabled	Enabled	Enabled
Run ActiveX controls and plugins	Enabled	Enabled. Potential candidate for risk acceptance.	Enabled
Script ActiveX controls marked safe for scripting	Enabled	Enabled. Potential candidate for risk acceptance.	Enabled
Submit non-encrypted form data	Enable/allow	Enable/allow	Enable/allow
Deny all add-ons unless specifically allowed in the Add-on List	See above in section on add-ons	See above in section on add-ons	n/a
Enable Native XMLHttpRequest Support	Enable/allow	Enable/allow. Potential candidate for risk acceptance.	n/a

6.10.8 A note on "Download signed/unsigned ActiveX controls"

During the evaluation phase none of the above listed sample websites (or other websites visited from test machines) showed any problems with a strict approach (that is: full deactivation/prohibition) concerning the download of signed/unsigned ActiveX controls. Given the discussions about these parameters in the company environment in the past it should be noted that the download of ActiveX controls (be them signed or unsigned) equivalents to the mostly (means: besides a potential check on malicious code by the MCP in place) *uncontrolled introduction* of (binary) code into the company´s environment. This in turn should be strictly prohibited. Furthermore it should be noted that even signed ActiveX controls by seemingly innocuous vendors (including security vendors) can turn out to include severe vulnerabilities *after* their deployment.

As an example the reader might have a look at the MS security advisory MS10-008:



⊟ Third-Party Kill Bits

This update includes kill bits to prevent the following ActiveX controls from being run in Internet Explorer:

- Symantec has requested a kill bit for the ActiveX control for Symantec WinFax Pro 10.3. Please see the post for more information about Symantec customers to set the kill bit for the control. If you have any questions or concerns regarding the kill bit for the Symantec WinFax Pro 10.3 control, Security at secure@symantec.com. The class identifier (CLSID) for this ActiveX control is:
  - {C05A1FBC-1413-11D1-B05F-00805F4945F6}.
- Google has requested a kill bit for the ActiveX control for Google Desktop Gadget v5.8. If you have any questions or concerns regarding Google D v5.8, please contact Google security at security@google.com. The class identifier (CLSID) for this ActiveX control is:
  - {5D80A6D1-B500-47DA-82B8-EB9875F85B4D}.
- Facebook has requested a kill bit for the ActiveX control for Facebook Photo Updater 5.5.8. The new control can be found here. If you have any qu regarding the kill bit for the Facebook Image Uploader ActiveX control, please contact Facebook Security. The class identifier (CLSID) for this Activ
  - {0CCA191D-13A6-4E29-B746-314DEE697D83}.
- Panda Security has requested a kill bit for the ActiveX control for PandaActiveScan Installer 2.0. The new control can be found here. If you have a regarding the kill bit for the Panda Security ActiveX control, please contact Panda Security. The class identifier (CLSID) for this ActiveX control is:
  - {2d8ed06d-3c30-438b-96ae-4d110fdc1fb8}.

6.11 Relevant parameters only existent in HKCU

One parameter was identified that presumably can't be set on a machine level: *Turn off inline AutoComplete in Windows Explorer*.

This feature allows Internet Explorer to remember form entry data for later retrieval and submission. It is also possible to store username(s) and password(s) with this feature. More importantly as for a security discussion, this feature is unable to discriminate between highly sensitive data and presumably publicly accessible information.

It is recommended to disable the *autocomplete* feature for forms and preventing auto completion for user names and passwords on forms. [see MS_IE8_SEC_GUIDE]

6.12 File Downloads

Downloading different types of files might both induce security problems and be necessary for a number of corporate internet uses at the same time. Subsequently this is an area of heavy discussion in many environments.

For the purpose of this document some flavors of downloads are distinguished, with regard to their security and usability impact. These are the following:

- ❑ Downloads of *data* files that means files to be processed by some type of locally installed application and not meant to be executed[2] by themselves. Typical examples include .PDF, .DOC or .XLS files. It should be noted that a large percentage of current targeted attacks works by means of such files (albeit then usually distributed by email). On the other hand all types of websites (including some of the sample set used here) provide some functionality based on file downloads, be it the reception of travel tickets, be it viewing account statements. Therefore, a general prohibition of file downloads would most probably cause quite some business impact/user annoyance followed by requests to add sites to the "trusted sites" zone (in case downloads are allowed in that one).
- ❑ Downloads of *executables* intended either to be *installed* (think: setup.exe) or to be *executed* – without *installation* process – directly (e.g. *Foxit Reader Portable*). Evidently both types of files induce heavy security risks and thus should be avoided at all[3]. Unfortunately the zone-based parameter that IE8 uses to control download behavior ("Allow file downloads", entry 1803) does *not* allow to differentiate between different types of files (e.g. based on their MIME type). The prohibition of this type of files should therefore be performed by some MCP entity.
- ❑ Downloads of binary files to be executed directly (potentially without any user interaction) in the browser context. For example this applies to add-ons like ActiveX controls, see discussion on this above.

Overall the following settings are recommended by the authors of this paper

Parameter	**Recommended setting for internet zone with MCP**	**Recommended setting for internet zone without MCP**	**Recommended setting for trusted sites**
Allow file downloads (applying to *data* files.) Note: IE does not allow the distinction used in this table, but the MCP entities probably will.	*Allow*. Limit types (based on MIME type or file extension) and/or scan for malicious code on MCP entities.	Deny. Security risk easily out-weighs busin. impact/user annoyance.	*Allow*. Limit types (based on MIME type or file exten-sion) and/or scan for malicious code on MCP.
Allow file downloads (applying to *executable* files)	*Allow*. Where possible deny on MCP.	Deny	*Allow*. Where possible deny on MCP.
Allow font downloads	Deny	Deny	Deny

[2] Execution *in the sense used here includes* installation *as well (so – obviously – these files are not meant to be* installed *either).*

[3] *It should be noted that the latter type of files (the to-be-executed-directly ones) at times are downloaded by users explicitly to circumvent security restrictions present on their systems.*

6.13 Cookies

6.13.1 Cookie Handling in Internet Zone

The treatment of cookies can be configured via the "Internet Options – Privacy" slider (in Internet Explorer's GUI). According to [MS_IE_ADVSettings] there's a complex interaction of a number of (registry) parameters once the settings on the "Internet Options – Privacy" slider are modified. Subsequently there seems no easily manageable way of deploying/enforcing the handling of cookies by registry settings (.reg-Files). Still setting the REG_DWORD entry "1A10" to "1" is a mandatory step for a number of settings, see [MS_IE_ADVSettings].

Furthermore for all sites in the "Intranet zone" and "Trusted sites" all cookies are usually accepted anyway, so modifying these settings will not prevent stealing cookies from intranet applications (which – from an attacker's point of view – is the most interesting scenario).

While the sample set of websites used here (see above) seem to work with a more strict setting than "Medium High" it should be noted that the authors expect problems with several websites in case of a "High" or "Block All Cookies" setting (e.g. logging into *ebay* requires a "Medium High" or lower setting).

From an overall "risk reduction while preserving usability" perspective "Medium High" seems the best option.

6.13.2 InPrivateBrowsing

To provide the possibility to use *InPrivateBrowsing* (a special feature of a given browsing session that deletes any cookies, temporary internet files, history etc. when the session is closed), the following parameter has to be set to "1" (REG_DWORD):

HKLM\Software\Policies\Microsoft\Internet Explorer\Privacy\EnableInPrivateBrowsing.

Still any user would have to invoke *InPrivateBrowsing* sessions manually (by navigating to the *Safety* menu) which means that no (behavior/security) enforcement can be performed by this parameter.

7 SECURITY CONSIDERATIONS FOR COMMONLY USED ADD-ONS

This chapter provides a security discussion and derived recommendations for a number of popular add-ons.

7.1 Adobe Reader

7.1.1 Overview & Security Aspects

Adobe Reader is able to provide the full feature set of PDF document functionality that includes simple documents, as well as interactive formulas, combined with DRM like security controls which can be defined within each document. Scripting of actions (e.g. verification of formula fields) is possible through the integration of JavaScript. As of Adobe Reader 8, functionality can be extended using self-written Plug-ins.

Adobe Reader's security features and settings are documented within [AD_APP_SEC_ACROBAT]. An overview of Adobe Reader related security bulletins is available from http://www.adobe.com/support/security/#readerwin.

Adobe reader has a long history of security vulnerabilities and is deemed responsible for a number of high profile targeted attacks. Security-wise not using it would certainly be desirable, but this is not regarded as a viable option for the vast majority of corporate environments.

7.1.2 On JavaScript in Adobe Reader

JavaScript displays one of the main threats for malicious code execution. JavaScript can be used to execute complex attacks scenarios, e.g. requiring certain memory areas filled with special data. Attack scenarios using JavaScript within Adobe Reader must be regarded as versatile as within Internet Explorer.

JavaScript execution is bundled to a verification mechanism to check for usage of High privileged (HP) API functions. By defining blacklists administrators could potentially prevent unknown JavaScript code from using HP functions. Adobe Reader's JavaScript library is frequently subject to vulnerabilities that allow execution of malicious code from within manipulated documents.

7.1.3 Recommendations

It should be evaluated if a deactivation of JavaScript in Adobe Acrobat Reader can be enforced. In the interim it can allowed on a document basis when opening a document which facilitates an approach of general disallowance with certain exceptions.

7.2 Adobe Flash

Adobe Flash (formerly Macromedia Flash) is a multimedia platform for active content especially used (mostly) in websites. Adobe Flash has a long history of security vulnerabilities. In 2009 five security bulletins for Flash 10 were released; each of them patches multiple security vulnerabilities. All of them were ranked as critical. Flash is mostly used as a browser plugin and therefore vulnerable to malicious content hosted on a website. Adobe Flash security vulnerabilities are widely exploited by drive-by-downloads.

For more detailed exploration of a secure Adobe Flash configuration consider the following two links:

http://www.adobe.com/devnet/flashplayer/articles/flash_player_admin_guide/flash_player_admin_guide.pdf

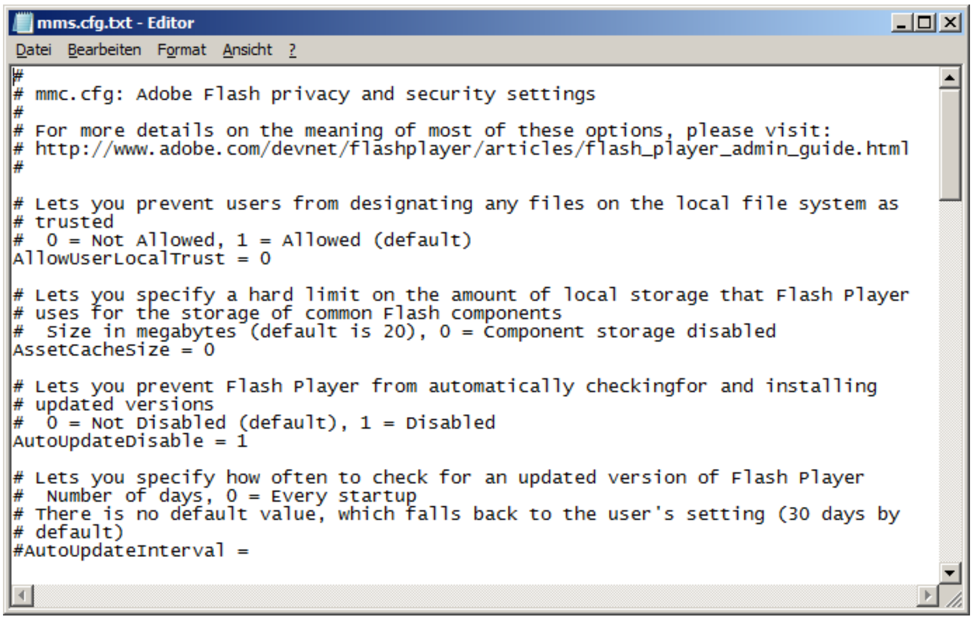
http://www.adobe.com/devnet/flashplayer/articles/flash_player_10_security.pdf

7.2.1 Hardening Adobe Flash

Flash can include many different file formats, for example sound, images, FLV (Flash Player compressed video format) and XML. For every environment where Flash is used it should be evaluated what kind of configuration best fits the needs while minimizing the functionality and therefore the risk of the Adobe Flash player. The security and privacy settings for Adobe Flash can be configured in the "mms.cfg" file which is located in the following directory (on Win XP):

"%WINDIR%\System32\Macromed\Flash"

The mms.cfg text file can be seen in the following screenshot:



```
#
# mmc.cfg: Adobe Flash privacy and security settings
#
# For more details on the meaning of most of these options, please visit:
# http://www.adobe.com/devnet/flashplayer/articles/flash_player_admin_guide.html
#

# Lets you prevent users from designating any files on the local file system as
# trusted
#   0 = Not Allowed, 1 = Allowed (default)
AllowUserLocalTrust = 0

# Lets you specify a hard limit on the amount of local storage that Flash Player
# uses for the storage of common Flash components
#   Size in megabytes (default is 20), 0 = Component storage disabled
AssetCacheSize = 0

# Lets you prevent Flash Player from automatically checkingfor and installing
# updated versions
#   0 = Not Disabled (default), 1 = Disabled
AutoUpdateDisable = 1

# Lets you specify how often to check for an updated version of Flash Player
#   Number of days, 0 = Every startup
# There is no default value, which falls back to the user's setting (30 days by
# default)
#AutoUpdateInterval =
```

More detailed information about the available settings in the mms.cfg text file can be found in the following file:

http://www.adobe.com/devnet/flashplayer/articles/flash_player_admin_guide/flash_player_admin_guide.pdf

7.2.2 Recommendations

Given the horrible security aspects of Flash and the low functionality impact of disabling Flash (in particular as for the sample websites used in this project) it is **strongly recommended** not to use Flash at all. The authors are well aware that this might cause some annoyance within the user community. Still the large security benefit easily outweighs the usability impact (mainly to be observed in websites of presumably "private use" anyway).

Furthermore approaches like *Blitzableiter* (http://blitzableiter.recurity.com/) should be closely followed if a corporate need for Flash is seen.

In case of Flash deployed on corporate workstations an associated *risk acceptance* must be filed.

It should be noted that "hardening Flash" as described above overall does only add minimal protection (as Flash vulnerabilities/exploits are based on architectural and parser problems).

7.3 Sun JAVA Virtual Machine / Runtime Environment

7.3.1 Overview & Security Aspects

Sun Java JRE is the runtime environment for the Java platform of Sun Microsystems. The Sun JRE includes the Java Virtual Machine and an API which provides the core classes and libraries of Java, like "*java.lang.string*", to run applications and applets written in Java. Sun Java VM has a long history of critical security vulnerabilities since several years and is constantly updated to close those vulnerabilities. In 2009, according to a major vulnerability database (at securityfocus.com), Sun Java JRE had 18 vulnerability entries, and some of them fix multiple critical security vulnerabilities.

7.3.2 DEP for Sun JAVA VM

Some years ago enabling DEP for Sun Java VM caused a number of problems and was subsequently cited as a major example of "DEP breaking things". However with JRE 6 Update 12 (http://bugs.sun.com/view_bug.do?bug_id=6674383) this has changed.

7.3.3 Recommendations

Subsequently it is recommended to enable DEP for *all programs or services* and not to add IE or (somehow) the JAVA VM to an exception list.

7.4 Apple QuickTime

7.4.1 Overview & Security Aspects

Quicktime is a proprietary framework developed from Apple capable of handling various formats of digital video, media clips, sounds and music. Quicktime Player, which is available for Apple Mac OSX and Windows platforms, has a long history of various critical security vulnerabilities in the past years. The latest Quicktime Player Version for example is prone to a buffer overflow vulnerability caused by malformed .mov files, which could lead to arbitrary code execution. All Quicktime Player versions are affected by this vulnerability and currently (as of 01/2010) there is no patch available from Apple.

7.4.2 Hardening Apple QuickTime

Even though Apple implemented additional security features (beginning with Version 7.4.5) such as ASLR (address space layout randomization), stack buffer safety checking and function call hardening, some of these features like ASLR can only be used on Windows Vista and newer operating systems from Microsoft. In order to reduce the attack surface, the Quicktime ActiveX Plugin in Internet Explorer should be disabled if it is not needed.

7.4.3 The role of the iPhone

In order to do several tasks with the iPhone, e.g. operating system updates or creation of backups, it is necessary to install iTunes. Unfortunately iTunes always includes Quicktime. QuickTime is a required component for iTunes to work properly as QuickTime provides iTunes with playback and encoding functionality, otherwise it may crash or a user might receive an error message indicating that Quicktime needs to be installed in order to run iTunes.

7.4.4 Recommendations

It is strongly recommended to evaluate if Quicktime or iTunes is needed on corporate machines at all. If both are not needed it is (obviously) recommended to not install any of these. If iTunes is only needed for iPhone use, the Quicktime ActiveX Plugin should be disabled.

7.5 Microsoft Silverlight

7.5.1 Overview & Security Aspects

Microsoft Silverlight is a web application framework that provides functionalities similar to those in Adobe Flash, integrating multimedia, graphics, animations and interactivity into a single runtime environment. Initially released as a video streaming plugin, later versions brought additional interactivity features and support for .NET languages and development tools. The current version, 3.0, was released on July 9, 2009, and version 4 beta was released on November 19, 2009[4].

It is compatible with multiple web browser products used on Windows, Linux and Mac OS X. To support and be compatible with Linux, FreeBSD and other open source platforms, Microsoft developed a free software implementation named Moonlight in cooperation with Novell.

Details of the Silverlight Security Model can be found here:

http://blogs.msdn.com/shawnfa/archive/2007/05/09/the-silverlight-security-model.aspx

http://blogs.msdn.com/shawnfa/archive/2007/05/10/silverlight-security-ii-what-makes-a-method-critical.aspx

http://blogs.msdn.com/shawnfa/archive/2007/05/11/silverlight-security-iii-inheritance.aspx

As of February 2010 Silverlight has had only one known security vulnerability (since the initial release) which could lead to remote code execution.

The details of the vulnerability can be viewed in the following Links:

http://www.microsoft.com/technet/security/bulletin/MS09-061.mspx

http://blogs.technet.com/srd/archive/2009/10/12/ms09-061-more-information-on-the-net-security-bulletin.aspx

7.5.2 Hardening Microsoft Silverlight

Currently the authors are not aware of any additional hardening steps for Silverlight on the client side.

7.5.3 Recommendations

It should be carefully evaluated if the Silverlight Plugin is needed. Obviously if Silverlight is not needed to deliver different types of media, it shouldn't be installed, in order to reduce the attack surface on the client machines. From a security point of view, Silverlight suffers from much less security vulnerabilities than Adobe Flash. If there is a choice how some content should be delivered/processed to/by the client browser (Silverlight or Flash) Silverlight might be the more reasonable approach.

[4] *Some of this information was taken from the Wikipedia article on Silverlight. It can serve as a point of reference if additional information is needed.*

8 Appendix A: Technical Details of Test Environment (x86)

The test environment consisted of a VMware ESX Server with a Windows XP virtual machine which was used during the test period. Initially both, that is a x86 and a x64 based platform, were used. After it turned out that the security aspects are mostly the same (except for some add-ons not running on 64-bit) subsequent tests were performed only on the x86 platform.

8.1 Operating System

On the virtual machine was a Windows XP installed with Service Pack 3 (Service Pack 2 for the XP x64) and Internet Explorer 8. All at the time of testing available Microsoft Patches were installed on the machine.

8.1.1 Installed Patches

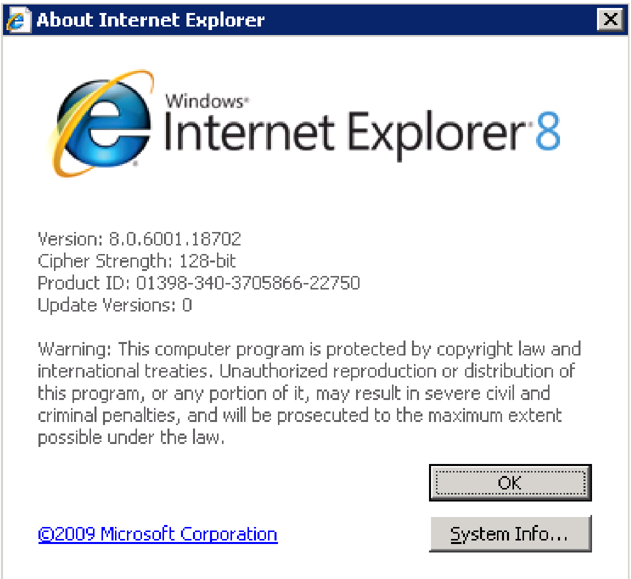
The following Patches were applied on the test system:

Security Update for Windows Internet Explorer 8 (KB976325)
Windows XP Service Pack 3 Service Pack KB936929
Security Update for Windows XP (KB923561)
Security Update for Windows XP (KB946648)
Security Update for Windows XP (KB950762)
Security Update for Windows XP (KB950974)
Security Update for Windows XP (KB951066)
Security Update for Windows XP (KB951376-v2)
Security Update for Windows XP (KB951748)
Update for Windows XP (KB951978)
Security Update for Windows XP (KB952004)
Hotfix for Windows XP (KB952287)
Security Update for Windows XP (KB952954)
Security Update for Windows XP (KB954459)
Security Update for Windows XP (KB955069)
Update for Windows XP (KB955759)
Security Update for Windows XP (KB956572)
Security Update for Windows XP (KB956744)
Security Update for Windows XP (KB956802)
Security Update for Windows XP (KB956803)
Security Update for Windows XP (KB956844)
Security Update for Windows XP (KB957097)
Security Update for Windows XP (KB958644)
Security Update for Windows XP (KB958687)
Security Update for Windows XP (KB958869)
Security Update for Windows XP (KB959426)
Security Update for Windows XP (KB960225)
Security Update for Windows XP (KB960803)
Security Update for Windows XP (KB960859)
Security Update for Windows XP (KB961371-v2)
Security Update for Windows XP (KB961501)

Update for Windows XP (KB967715)
Update for Windows XP (KB968389)
Security Update for Windows XP (KB969059)
Security Update for Windows XP (KB969947)
Security Update for Windows XP (KB970238)
Security Update for Windows XP (KB970430)
Security Update for Windows XP (KB971486)
Security Update for Windows XP (KB971557)
Security Update for Windows XP (KB971633)
Security Update for Windows XP (KB971657)
Update for Windows XP (KB971737)
Security Update for Windows XP (KB971961)
Security Update for Windows XP (KB973354)
Security Update for Windows XP (KB973507)
Security Update for Windows XP (KB973525)
Update for Windows XP (KB973687)
Update for Windows XP (KB973815)
Security Update for Windows XP (KB973869)
Security Update for Windows XP (KB973904)
Security Update for Windows XP (KB974112)
Security Update for Windows XP (KB974318)
Security Update for Windows XP (KB974392)
Security Update for Windows XP (KB974571)
Security Update for Windows XP (KB975025)
Security Update for Windows XP (KB975467)
Hotfix for Windows XP (KB976098-v2)
Security Update for Windows XP (KB976325)

8.1.2 Internet Browser

Internet Explorer 8 was installed on the test system and all IE8 Patches were applied.



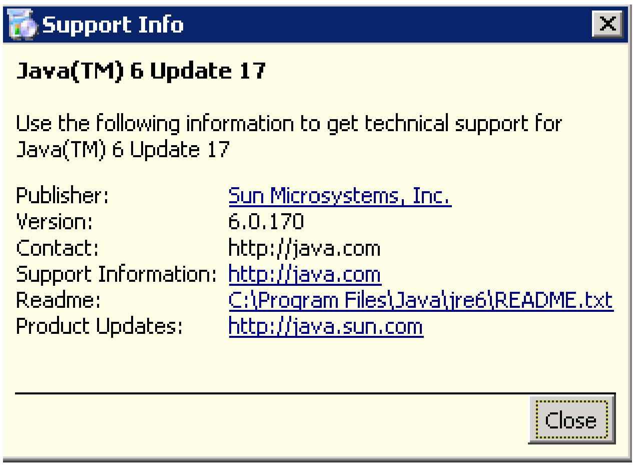
Version: 8.0.6001.18702
Cipher Strength: 128-bit
Product ID: 01398-340-3705866-22750
Update Versions: 0

Warning: This computer program is protected by copyright law and international treaties. Unauthorized reproduction or distribution of this program, or any portion of it, may result in severe civil and criminal penalties, and will be prosecuted to the maximum extent possible under the law.

©2009 Microsoft Corporation

8.1.3 Java Runtime Environment

Java 6 Update 17 was installed on the test system.



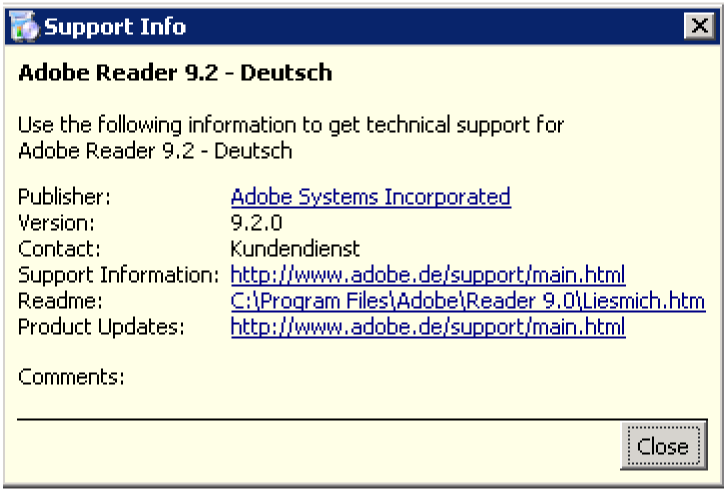
Java(TM) 6 Update 17

Use the following information to get technical support for Java(TM) 6 Update 17

Publisher:	Sun Microsystems, Inc.
Version:	6.0.170
Contact:	http://java.com
Support Information:	http://java.com
Readme:	C:\Program Files\Java\jre6\README.txt
Product Updates:	http://java.sun.com

8.1.4 Adobe Acrobat Reader

Adobe Acrobat Reader 9.2 was installed on the test system.



Adobe Reader 9.2 - Deutsch

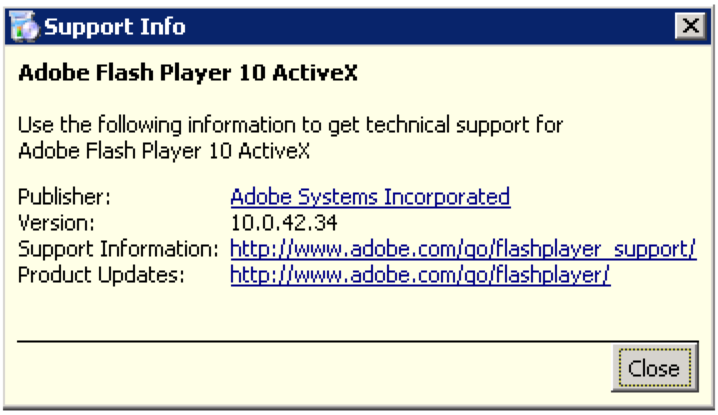
Use the following information to get technical support for Adobe Reader 9.2 - Deutsch

Publisher:	Adobe Systems Incorporated
Version:	9.2.0
Contact:	Kundendienst
Support Information:	http://www.adobe.de/support/main.html
Readme:	C:\Program Files\Adobe\Reader 9.0\Liesmich.htm
Product Updates:	http://www.adobe.de/support/main.html

Comments:

8.1.5 Adobe Flash Plugin

Adobe Flash Player 10 ActiveX Version 10.0.42.34 was installed on the test system.



Support Info

Adobe Flash Player 10 ActiveX

Use the following information to get technical support for
Adobe Flash Player 10 ActiveX

Publisher:	Adobe Systems Incorporated
Version:	10.0.42.34
Support Information:	http://www.adobe.com/go/flashplayer_support/
Product Updates:	http://www.adobe.com/go/flashplayer/

9 AppenDix B: Regfile listing Default Add-ONs

In the following the CLSIDs of add-ons being present after performing a "fresh installation of the overall environment" (for details on test environment see above) are listed, as a "regfile":

```
Windows Registry Editor Version 5.00

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\policies\Ext]
"NoFirsttimeprompt"=dword:00000001
"RestrictToList"=dword:00000001
"ListBox_Support_CLSID"=dword:00000001

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\policies\Ext\CLSI
D]
```

Microsoft Java VM:
```
"{08B0E5C0-4FCB-11CF-AAA5-00401C608501}"="1"
```

InformationCardSigninHelper Class:
```
"{19916E01-B44E-4E31-94A4-4696DF46157B}"="1"
```

XML DOM Document Microsoft.XMLDOM.1.0:
```
"{2933BF90-7B36-11D2-B20E-00C04F983E60}"="1"
```

XSL Template Msxml2.XSLTemplate:
```
"{2933BF94-7B36-11D2-B20E-00C04F983E60}"="1"
```

HtmlDlgSafeHelper Class :
```
"{3050F819-98B5-11CF-BB82-00AA00BDCE0B}"="1"
```

Tabular Data Control:
```
"{333C7BC4-460F-11D0-BC04-0080C7055A83}"="1"
```

XML Schema Cache:
```
"{373984C9-B845-449B-91E7-45AC83036ADE}"="1"
```

JavaWebStart:
```
"{5852F5ED-8BF4-11D4-A245-0080C6F74284}"="1"
```

Microsoft Shell UI Helper:
```
"{64AB4BB7-111E-11D1-8F79-00C04FC2FBE1}"="1"
```

Windows Media Player:
```
"{6BF52A52-394A-11D3-B153-00C04F79FAA6}"="1"
```

Microsoft Update Web Control Class:
```
"{6E32070A-766D-4EE6-879C-DC1FA91D2FC3}"="1"
```

XML DOM Document 6.0:
```
"{88D96A05-F192-11D4-A65F-0040963251E5}"="1"
```

Free Threaded XML DOM Document 6.0:

```
"{88D96A06-F192-11D4-A65F-0040963251E5}"="1"
```

XML Schema Cache 6.0:

```
"{88D96A07-F192-11D4-A65F-0040963251E5}"="1"
```

XSL Template 6.0:

```
"{88D96A08-F192-11D4-A65F-0040963251E5}"="1"
```

XML HTTP 6.0

```
"{88D96A0A-F192-11D4-A65F-0040963251E5}"="1"
```

Java Plug-in 1.6.0_17:

```
"{8AD9C840-044E-11D1-B3E9-00805F499D93}"="1"
```

VB Script Language:

```
"{B54F3741-5B07-11cf-A4B0-00AA004A55E8}"="1"
```

Java Plug-in 1.6.0_17:

```
"{CAFEEFAC-0016-0000-0017-ABCDEFFEDCBA}"="1"
"{CAFEEFAC-0016-0000-0017-ABCDEFFEDCBB}"="1"
"{CAFEEFAC-0016-0000-0017-ABCDEFFEDCBC}"="1"
"{CAFEEFAC-FFFF-FFFF-FFFF-ABCDEFFEDCBA}"="1"
```

Java Deployment Toolkit Plugin:

```
"{CAFEEFAC-DEC7-0000-0000-ABCDEFFEDCBA}"="1"
```

Adobe Shockwave Flash Object:

```
"{D27CDB6E-AE6D-11CF-96B8-444553540000}"="1"
```

Java Plug-In 2 SSV Helper:

```
"{DBC80044-A445-435B-BC74-9C25C1C588A9}"="1"
```

Adobe getPlus Helper Plugin:

```
"{E2883E8F-472F-4FB0-9522-AC9BF37916A7}"="1"
```

Diagnose Connection Problems:

```
"{E2E2DD38-D088-4134-82B7-F2BA38496583}"="1"
```

JQSIEStartDetectorImpl Class:

```
"{E7E6F031-17CE-4C07-BC86-EABFE594F69C}"="1"
```

Scripting.Dictionary VB Object:

```
"{EE09B103-97E0-11CF-978F-00A02463E06F}"="1"
```

JScript Language:

```
"{F414C260-6AC0-11CF-B6D1-00AA00BBBB58}"="1"
```

MSXML Parser 3.0 .cab file redistribution:

```
"{F5078F32-C551-11D3-89B9-0000F81FE221}"="1"
```

MSXML2.FreeThreadedDOMDocument.3.0:

```
"{F5078F33-C551-11D3-89B9-0000F81FE221}"="1"
```

MSXML2.XMLSchemaCache.3.0:

```
"{F5078F34-C551-11D3-89B9-0000F81FE221}"="1"
```

MSXML2.XMLHTTP.3.0:

```
"{F5078F35-C551-11D3-89B9-0000F81FE221}"="1"
```

MSXML2.XSLTemplate.3.0:

```
"{F5078F36-C551-11D3-89B9-0000F81FE221}"="1"
```

MSXML2.DSOControl.3.0:

```
"{F5078F39-C551-11D3-89B9-0000F81FE221}"="1"
```

MSXML2.FreeThreadedDOMDocument:

```
"{F6D90F12-9C73-11D3-B32E-00C04F990BB4}"="1"
```

MSXML2.DSOControl:

```
"{F6D90F14-9C73-11D3-B32E-00C04F990BB4}"="1"
```

Windows Messenger:

```
"{FB5F1910-F110-11D2-BB9E-00C04F795683}"="1"
```

10 Appendix C : Results of Some Security Checks with Template Applied

heise Security Browsercheck [as of 01/10/2010]

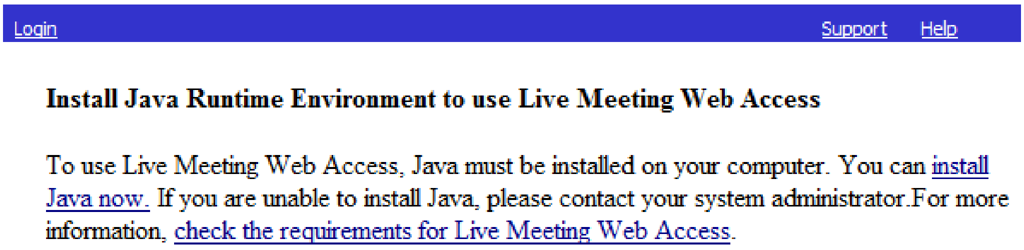
http://www.heise.de/security/dienste/Demos-fuer-den-Internet-Explorer-437526.html

No vulnerabilities found.

11 APPENDIX D: VARIOUS KNOW PROBLEMS

11.1 JAVA has to be re-installed after modifying add-on behaviour

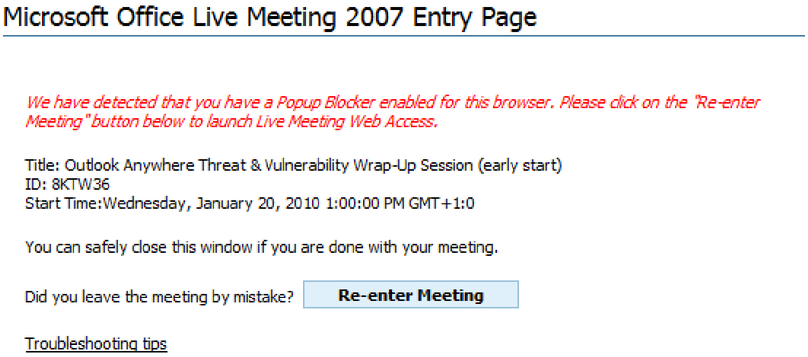
After modifying the add-on behaviour (and even after going with a "liberal approach" again), MS LiveMeeting (working properly before the parameter change) could not be started:



Install Java Runtime Environment to use Live Meeting Web Access

To use Live Meeting Web Access, Java must be installed on your computer. You can install Java now. If you are unable to install Java, please contact your system administrator.For more information, check the requirements for Live Meeting Web Access.

After re-installing JAVA the following message appeared (based on other parameters) and subsequent use of LiveMeeting was possible:



Microsoft Office Live Meeting 2007 Entry Page

We have detected that you have a Popup Blocker enabled for this browser. Please click on the "Re-enter Meeting" button below to launch Live Meeting Web Access.

Title: Outlook Anywhere Threat & Vulnerability Wrap-Up Session (early start)
ID: 8KTW36
Start Time:Wednesday, January 20, 2010 1:00:00 PM GMT+1:0

You can safely close this window if you are done with your meeting.

Did you leave the meeting by mistake?

Questions concerning this topic as well as related ones may be directed to us. We will be glad to support you.

In the name of our team, yours sincerely,

Friedwart Kuhn.

ERNW GmbH
Friedwart Kuhn
Senior Security Consultant
fkuhn@ernw.de

ERNW Enno Rey Netzwerke GmbH
Breslauer Str. 28
69124 Heidelberg
Tel. +49 6221 480390
Fax +49 6221 419008
Mobil +49 15152411855
www.ernw.de