Global Information Assurance Certification Paper

Malicious Mobile Code

GIAC Security Essentials Certification (GSEC)
Practical Assignment
Version 1.4c

Option 1

Submitted 9th December 2004.

Daniel Cleary

Malicious Mobile Code

Introduction:

This document will explore some of the most common threats posed by malicious mobile code from the internet to an organisation's IT infrastructure. This document focuses on how mobile code can be used to attack a host through the most widely used browser Microsoft's Internet Explorer.

Mobile Code is a lightweight program that is downloaded from a remote system and executed locally with minimal or no user intervention. [1]

Organisations spend a lot of money on products such as firewalls to protect their infrastructure from threats on the internet. However in order for everyday business to continue users within an organisation must have access to browse web sites; so a hole is punched through the firewall to facilitate this requirement. Internet browsers have evolved a lot, from their initial conception, when they just allowed web pages to be presented in plain HTML. Active content has been added to web sites to present a more dynamic and richer experience. However this opens up avenues of attack to nefarious individuals. SANS have rated this threat as number 6 in The Twenty Most Critical Internet Security Vulnerabilities. [2] Throughout this document suggestions are made to prevent and/or mitigate against the various threats. Where the method of protection is specific to a particular type of mobile code I have listed it in its respective section. General steps that can be taken to reduce exposure to malicious mobile code have been provided at the end.

Most common types of mobile code:

Scripting Languages

The two most common scripting languages are JavaScript and VB Script.

JavaScript is a general purpose scripting language, originally developed by Netscape, whose code can be embedded within standard Web Pages to create interactive documents. JavaScript has nothing to do with Java, apart from a similar syntax. Netscape created JavaScript and named it so for marketing reasons. Microsoft has a variation of JavaScript called Jscript. JavaScript is

supported in a multitude of browsers including Internet Explorer, Mozilla Firefox, and Netscape Navigator.

VBScript is a scripting language that was developed by Microsoft. It is only supported within Internet Explorer.

Scripts are automatically loaded by browsers without a request being made to the end user; if active scripting is not switched off. Some of the most common attacks by scripting languages include:

Resource Exhaustion: WinBomb is one example of this. It attempts to open an unlimited number of browser windows. This does not pose a serious threat to users but it is never the less an inconvenience.

Browser Hijacking: This is more of a threat than resource exhaustion since the control of the web browser is taken out of the hands of the user. Examples of this include not being allowed to leave a site because browsers pop-up that are larger than a user's display or redirection to another site.

Cookie Stealing: Cookies can often store sensitive date. They can be non-persistent, deleted on a browsers exit, or persistent, expires at a future time. It can be used to maintain a state since HTTP is stateless, for example, when you browse a website and it automatically loads your preferences. Of most interest to attackers are session identifiers (SID) stored in cookies because they could potentially be used to hijack a users session with a web site without the need for a username or password; depending on the authentication mechanism in place. [1]

XSS – Cross Site Scripting: Attacks make use of custom URL or code injection into a valid web-based application URL or imbedded data field. Some of the more high profile attacks using XSS include phising attacks, the process of tricking or socially engineering an organisations customers into imparting their confidential information for nefarious use. [3]

Another danger of these scripting languages is the way that they can be used to pass command to ActiveX components via VBScript and Java via JavaScript.

Active X

ActiveX controls are compiled components that can be used by multiple programs or Web sites. They can be written in any programming language that supports Microsoft's Component Object Model (COM), and they are implemented as .ocx or .dll files. [4] A number of ActiveX controls are included in Windows Operating Systems while others can be downloaded from the internet.

Trust is the security model used by Active X controls. If a user encounters an Active X control on a website an alert will be displayed explaining to the user what is happening and whether the Active X control is signed or not. A user must decide whether to trust the publisher or not. Microsoft advocates the use of Authenticode technology to authenticate the code and ensure its integrity. Authenticode involves purchasing an Authenticode Code Signing Certificate from a trusted CA such as Verisign and using the associated private key to sign the Active X control with. If a control has been digitally signed it assists a user in making a decision as to whether to install the Active X control.

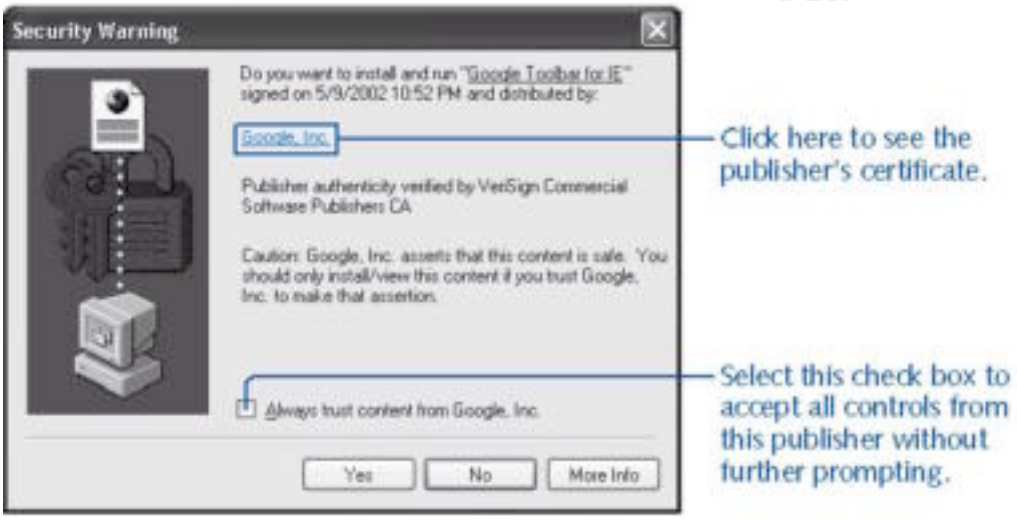


Figure 1- Active X Security Warning [4]

In figure 1 above you can click on the publisher's name to view the details of the code signing certificate. Also there is an option to always trust content from the publisher without ever being prompted again. Once a user clicks yes to install the component there are no permissions or safeguards on what it can do. The Active X control can now perform any action that the user can.

When using this model there is a heavy reliance on the reputation of the publisher that they would not cause any harm to your PC. Although code signing can guarantee the identity of the control author and guarantee that the control has not been tampered with, it does not guarantee that the code is free from errors and security vulnerabilities. [5]

Two particularly dangerous settings for ActiveX controls are:

Safe for Initialisation – control will do no harm regardless of how it's initialised
Safe for Scripting - control will do no harm, regardless of how its properties, events, and methods are used. [6]

These are very grand guarentees that developers have to accept. The first rule of safe for initialisation and scripting is this:

Your control is not safe for initialisation or safe for scripting [5]

Although there are a number of malicious ActiveX controls in the wild such as Exploder; which can shut down a PC. A person still has to accept an ActiveX control as shown in Fig. 1. An easier attack vector is exploiting existing legitimate ActiveX controls that have been badly programmed.

eEye Digital Security released a security vulnerability in IBM's signed "acpRunner" ActiveX on June 15th, 2004. IBM designed the ActiveX control to allow automated support for their PCs. Methods with the control included "DownLoadURL", "SaveFilePath" and "Download" which could be exploited for example to download a malicious executable to be automatically executed on start-up. [7]

The above IBM example is an example of a Non-Malicious ActiveX control being utilised in a way that it was not originally intended.

Java

Unlike ActiveX, Java is an actual programming language. Java is designed to run in two scenarios: java code that is on the local machine and java code that is downloaded from the network. I will only focus on applets which are small Java applications downloaded from a Web page to a browser and executed either by the browser's Java Virtual Machine or via a java plug-in.

One of the major advantages of Java is that Java applets are implemented within a sandbox. The sandbox is an area that untrusted Java code is allowed to run which constrains Java to conform to certain rules such as not allowing Java to read or write to or from the local files system.

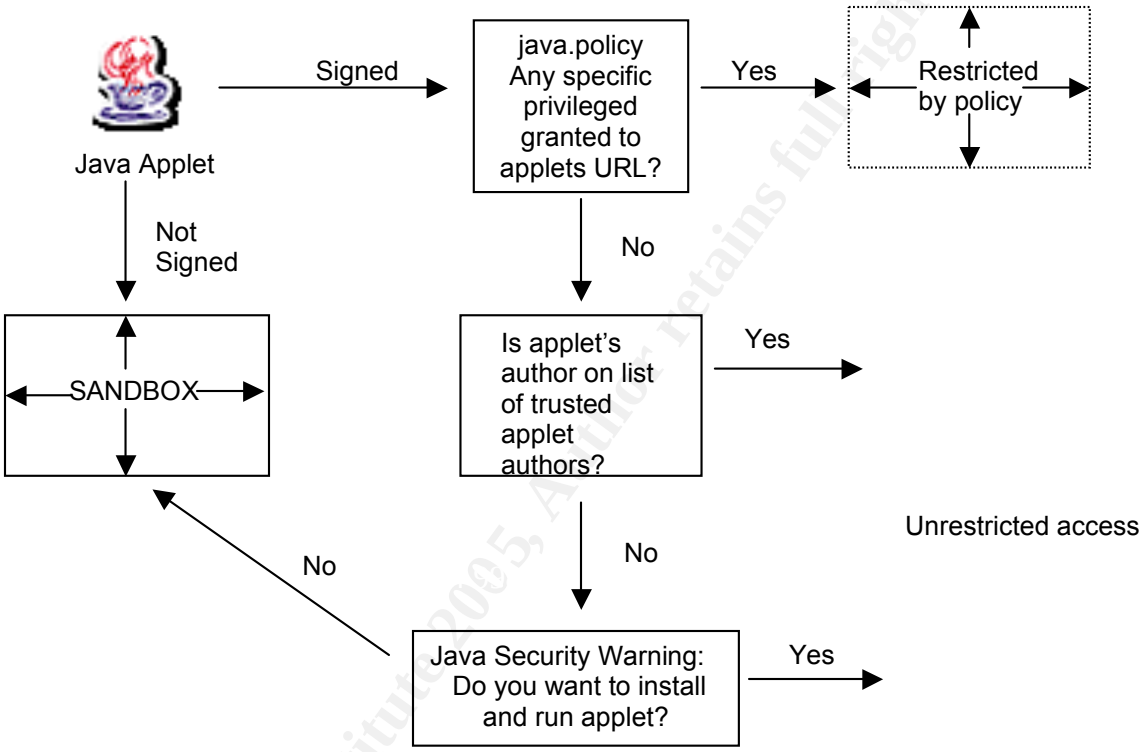
The methods by which applets are allowed to be executed have changed with the different versions of Java:

- In version 1.0 remote code was constrained only to the sandbox.
- JDK 1.1 – the previous model constraints were too severe. Java decided to introduce an ActiveX like Trust model that allowed the applet to be signed and if the end user decided to trust the signed applet it was not constrained by the sandbox and has full access to the host. A signed java applet consists of a signed JAR (Java Archive).

- Java 2 Security – Refined the previous Java security model again by adding the option of restricting a signed applets access from unrestricted access to restricted access as defined in the security policy. [8]

Please refer to the Appendix for a more detailed explanation of the Java2 Security architecture.

Below is a representation of how Applets are handled in the Java2 security Architecture:



Just to summarise a signed applet will have unrestricted access to a system if it is on the list of trusted applet authors or if the end user chooses to allow the applet to install and run. The applet author will be added to the list of trusted authors for future reference. If restricted access by the applet is desired then the java.policy file needs to be edited to grant specific privileges. This file is normally located in the (java.home)/lib/security/ directory.

There have been relatively few malicious applets compared to ActiveX controls. However there have been issues in how the Java has been implemented by web browsers.

In the last month Sun released the following vulnerability alert:

A vulnerability in the Java Plug-in may allow an untrusted applet to escalate privileges, through JavaScript calling into Java code, including reading and writing files with the privileges of the user running the applet. [9]

Plug-ins

Although plug-ins allow browsers to support new types of content, they are not active content in themselves, but simply executables that enable active content technologies [10]. Below is a list of common plug-ins with examples of vulnerabilities to illustrate some of the ways they can increase exposure to malicious mobile code.

Flash

Secunia issued the following advisory for Macromedia Flash:

Macromedia Flash does not respect the privacy settings in Internet Explorer (and other browsers) when playing content. This allows malicious sites to keep tracking users even if they have disabled other cookie functions and storing of user data in their browser and the "super-cookie" in Mediaplayer.

Solution:
If privacy is crucial to anonymise your users, we recommend that you un-install Macromedia Flash or filter Flash content via a proxy and only allow Flash from trusted sites. [11]

Shockwave

In November of 2002 Macromedia released new versions of all its Shockwave players to fix the following issue. Maliciously authored Shockwave content, working in conjunction with other content on a Web server, could read the contents of files from the local file systems of Shockwave Player users, and send those contents back to Web servers without users' consent or knowledge. This vulnerability was limited to files whose locations and names are known or guessed ahead of time by attackers. An attacker would have to entice the user to a site under his control to exploit this vulnerability. This vulnerability can't be used to modify or delete local files. [12]

Java Plug-In

There is a choice between what Java Virtual Machine can be used with Internet Explorer. Microsoft has it's own implementation of the JVM or alternatively you can use Sun's, actively supported, JVM via a Java Plug-In, installed as an ActiveX control, in IE. The most recent vulnerability associated with the Java

Plug-In is listed above in the Java section. Information on how to remove the MS JVM can be found at:http://www.winnetmag.com/Windows/Article/ArticleID/38206/Windows_38206.html

BHOs

.

Browser Helper Objects are plug-ins written specifically for IE. Currently spyware programs preferred platform is IE. [1] There are numerous anti-spyware products such as Spyware Blaster and Adaware to detect and remove spyware programs.

.NET

At a high-level, the .Net Framework gives developers and administrators granular security control over their applications and resources, eliminates many of the major security risks facing applications today due to flawed code, such as buffer overflows, and shifts the burden from having to make critical security decisions – such as whether or not to run a particular application or what resources that application should be able to access – from end users to developers and administrators. [13]

Basic components of the Framework:

- Common Language Runtime (CLR) – performs just in time JIT compilation. Runs and manages executing code. It enforces restrictions to keep the code from behaving unexpectedly.
- Class Libraries – used by developers to write programs to execute in CLR.
- Assemblies – executable or DLL compiled using one of .NET Framework's many language compliers. They contain code that the runtime executes in the form of MSIL (Microsoft Intermediate Language). They also contain Metadata used to locate and load classes, enforce security etc. Through assemblies the CLR and class libraries implement the managed code architecture of the .NET Framework. [14]

There have only been a few vulnerabilities discovered in .Net to date.

Steps to Reduce the Risk of Malicious Mobile Code

The choice taken to mitigate will rely upon “Two main factors should be considered when attempting to gauge the risk involved with active content technologies: the capabilities or facilities of the programming language and the breadth and strength of controls in place within the execution environment to enforce policy.” [10] For example JavaScript is limited to what it can do within the browser. ActiveX alternatively has no controls in place once installed.

At the moment critical security decisions are being made by end users within organisations: Should I click yes to this Security Warning to install this Applet or ActiveX control? The following list contains a range of options that can be implemented for varying degrees of protection.

1. Education & Policy
   People often know that they are violating policy but do not understand the risk of their actions. Formulate an Acceptable Internet Usage Policy which categorises the risks associated with the different mobile code and ensure that all employees have read it. It should make the decision about whether to install an ActiveX control a lot easier.

2. Ensure all relevant patches and service packs have been applied
   This is applicable not only to the IE but also to the Operating System. The latest Service Pack for XP was SP2. This service pack introduced a number of new features such as the Manage Add-ons option to view and enable/disable plug-ins/ActiveX controls, previously it took a bit more effort to find and diable ActiveX Controls. Other additions included IE Window Restrictions to ensure that scripts cannot position windows so that the title bar or address bar are above the visible top of the display. For more information on the updates to IE in SP2 go to:
   http://www.microsoft.com/technet/prodtechnol/winxppro/maintain/sp2brows.mspx.htm

   Vulnerabilites are constantly being found in Internet Explorer so regular updates are required. To assist in this effort sign up for free mail notifications from Advisory companies such as Secunia. Also plug-in updates need to be monitored. In the case of the Java vunerability that was mentioned earlier the resolution to the issue was to apply the latest Java Virtual Machine to mitigate against this risk.

3. Use of anti-virus software with up to date virus definitions.
   This can help prevent against known attacks. Anti-virus software vendors are improving their ability to tackle malicious mobile code. McAfee VirusScan

Enterprise 8.0i is the industry's first anti-virus software to offer intursion prevention security with application-specific buffer overflow prevention.[15]

4. Security Zone Configuration
   The way in which IE handles active content can be controlled within the Security Settings in Internet Options. Security Zones are explained in detail in the Appendix with recommendations for configuration. If there is no requirement for ActiveX controls, scripting, Java within an organisation then they should be disabled as appropriate.

5. Take the choice away from the user. Do not allow a user to customise the Security Settings and lower the Security Zone Settings. Standardise the security settings for a domain using Microsoft's Internet Explorer Administration Kit IEAK from http://www.microsoft.com/windows/ieak/default.asp . Also the Group Policy Computer Configuration\Administrative Templates\Windows Components\ Internet Explorer settings for: Security Zones can be set to prevent any changes. [16]

6. Disable Automatic Install of Internet Explorer components and software restriction
   This prevents IE from downloading components when users browse to Web sites that require these components to fully function. This can be done in the Group Policy Computer Configuration\Administrative Templates\Windows Components\ Internet Explorer. Use Software Restriction Policies to protect against unauthorized software as outlined in http://support.microsoft.com/default.aspx?scid=kb;EN-US;322756

7. Use Mobile Code Scanning Solution to detect Malicous Code – Behaviour Monitoring.
   Applications such as Trend Interscan AppletTrap HTTP Proxy detect and prevent malicious Java Applets, Scripts and ActiveX Controls
   Finjan, the market leader, offers behaviour monitoring of mobile code that will either quarntine the mobile code if found to be mailicious or if it doesn't comply with policy .

8. Malicious website blocking
   Products such as Websense Enterprise Premium Group III, an add on database that identifies Web sites containg malicious mobile code, blocks access to Web sites that contain malicious mobile code. Updated list of malicious sites in nightly download from Websense. [17]

9. Restrict the rights of the end user.

Some organisations use a locked down workstation where regular users have limited rights on their PC. This can help prevent malicious mobile code attacks as they are run in the context of the user. Administrators should switch to a non-admin account when browing the internet.

10. Audit existing ActiveX components for vulnerabilities and disable them if not required.

Conculsion

Throughout this document the most common types of mobile code have been explored with a view to identifying the potential threats posed and steps to mitigate against the risk. Micorsoft's web browser, Internet Explorer, is the most sophisticated web browser available. Consequently it has a very long track record of vulnerability discovery and remote exploitation. For typical web browsing, less than 5% of it's built-in functionality is used. In fact many of the "features" available in the browser were added to protect against previous flaws and attack vectors. Unfortuately each new feature brings with it a host of security problems and additional complexity. [3]

Bibliography

[1] Skoudis, Ed and Zeltser, Lenny, Malware: Fighting Malicious Code. Prentice Hall, 2004.

[2] "The Twenty Most Critical Internet Security Vulnerabilities (Updated) ~ The Experts Consensus." SANS Top 20 List, 8 October 2004 http://www.sans.org/top20/

[3] Ollman, Gunter, "The Phishing Guide – Understanding & Preventing Phishing Attacks", 23 September 2004 http://www.nextgenss.com/papers/NISR-WP-Phishing.pdf

[4] Bott, Ed and Siechert, Carl, Microsoft Windows Security for Windows 2000 and XP – Inside Out. Microsoft Press, 2002.

[5] Howard, Michael and LeBlanc, David; Writing Secure Code 2nd Edition Microsoft Press, 2002.

[6] Finnegan, Sean, "Managing Mobile Code with Microsoft Technologies", Microsoft TechNet, 31 August 2000 http://www.microsoft.com/technet/security/bestprac/mblcode.mspx

[7] "IBM acpRunner Activex Dangerous Methods Vulnerability." eEye Digital Security Published Advisories, 15 June 2004 http://www.eeye.com/html/research/advisories/AD20040615A.html

[8] Srinivas, Raghavan N. "Java security evolution and concepts, Part 2" Java World, July 2000 http://www.javaworld.com/javaworld/jw-07-2000/jw-0728-security.html

[9] "Security Vulnerability With Java Plug-in in JRE/SDK" SunSolve, 2 December 2004 "http://sunsolve.sun.com/search/document.do?assetkey=1-26-57591-1&searchclause=%22category:security%22%20%22availability,%20security%22 http://www.idefense.com/application/poi/display?id=158&type=vulnerabilities

[10] Jansen, Wayne A," Guidelines on Active Content and Mobile Code" NIST Special Publications, October 2001 http://csrc.nist.gov/publications/nistpubs/800-28/sp800-28.pdf

[11] "Macromedia Flash bypasses cookie security" Secunia Advisories, 8 October 2002 http://secunia.com/advisories/7245/

[12] "MPSB02-11 - Macromedia Shockwave URL Modification Issue" Macromedia DevNet, November 2002

http://www.macromedia.com/devnet/security/security_zone/mpsb02-11.html
[13] “Security in the Microsoft .NET Framework” Foundstone Resources, Date Unavailable http://www.foundstone.com/resources/whitepapers/dotnet-security-framework.pdf

[14] “Code Access Security and Policy in Microsoft’s .NET” SANS Reading Room, 1 January 2003 http://www.sans.org/rr/whitepapers/windowsnet/976.php

[15] “McAfee VirusScan Enterprise 8.0i” McAfee System Protection Solutions, May 2004
http://www.mcafeesecurity.com/us/_tier2/products/_media/mcafee/ds_virusscan80i.pdf

[16] Microsoft “Using Software Restriction Policies to Protect Against Unauthorized Software” Microsoft TechNet, 1 January 2002
http://www.microsoft.com/technet/prodtechnol/winxppro/maintain/rstrplcy.mspx

[17] Sturdevant, Cameron. “Websense Checks Mobile Code” eWeek – The Enterprise News Weekly, 12 August 2002
http://www.websense.com/company/news/websense_eweek_review_08-02.pdf

[18] Taylor, Art, Brian Buege and Randy Layman Hacking Exposed – J2EE & Java Osborne/McGraw-Hill, September 2002.

Appendix

Java 2 Security

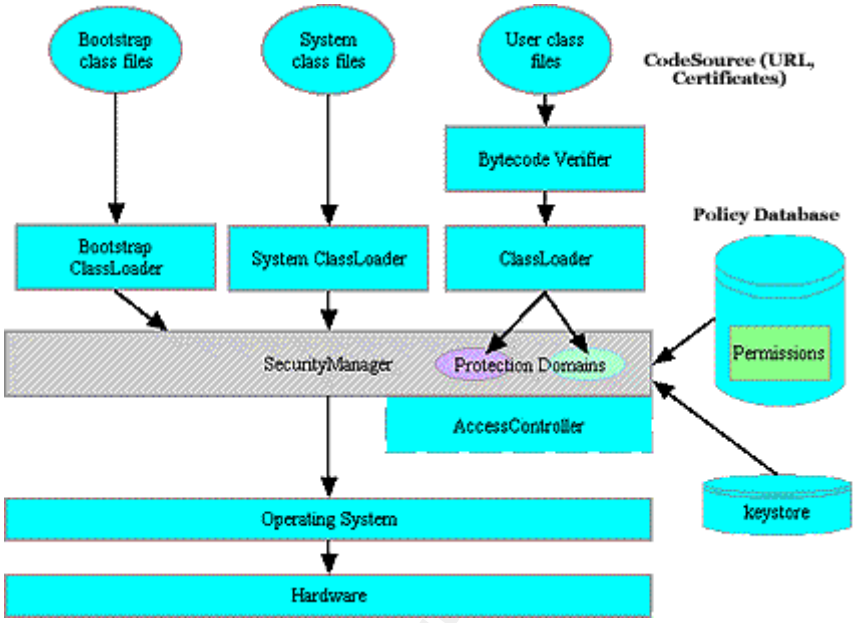


Figure 2 - Java 2 Security Architecture [8]

The Java runtime environment (JRE) executes a Java program by reading the bytecode in a class file and executing program instructions as they are encountered. The Bytecode Verifier performs various validation checks such as checking type and stack usage.

A Java *class loader* is responsible for loading the classes or bytecode that composes the program into the Java Virtual Machine (JVM). As you can see from Fig 2 above the there are two separate ClassLoaders within the Java2 Security Architecture. One is used for local environment and the other for classes being loaded over the network

Protection Domains – a domain is a portion of operating environment to which you will apply security. A domain conceptually groups sets of classes that have all been granted the same permissions. A *security policy* defines the *protection domain.* In a similar way to a sandbox; the protection domain model requires that the actions of classes executing within the protection domain have appropriate permissions to perform the actions they attempt or else an exception will be thrown. [18]

A protection domain is provided using a code source and associating permissions with that code source. Code being executed is associated with a

code source which identifies where the code was loaded from. It is critical to know the code's origin and author. Code Source is represented by a URL location either local or remote all any classes with same URL and optionally having the same signature and key will be placed in the same domain and will be granted the same permissions. The security policy manages the association between code source and permissions.

The *Security Manager,* as seen in Fig 2, is at the focal point of authorization and can perform a number of checks such as to check if read access is allowed to a file and check to see if the requested access has the given permission based on the policy etc.

The *Access Controller* is used for three purposes:

1. To decide whether access to a critical system resource should be allowed or denied, based on the security policy currently in effect
2. To mark code as privileged, thus affecting subsequent access determinations
3. To obtain a snapshot of the current calling context, so access-control decisions from a different context can be made with respect to the saved context [8]

The *Keystore* is a password-protected database that holds private keys and and digital certificates. The password is selected at the time of creation. Each database entry can be guarded by its own password for extra security. Certificates that come pre-installed, such as Verisign's, and certificates that are imported into the keystore are considered to be trusted. Keystore information can be used and updated by the security tools provided with the SDK.

Configuring Security Zones

Security Zones:
IE consists of 4 Security zones

- Local Intranet:- sites within within organisation
- Trusted Sites: - outside firewall but sites you have the highest trusts for e.g. business associates.
- Restricted:- sites you explicitly distrust
- Internet: any sites not in the above.

These are accessible through Tools -> Internet Options -> Security.

A fifth zone, not normally configurable, My Computer is used run ActiveX controls installed Windows and file URLs that reference local documents [4]

Below is a list of sites that are added to the Local Intranet Zone by default:

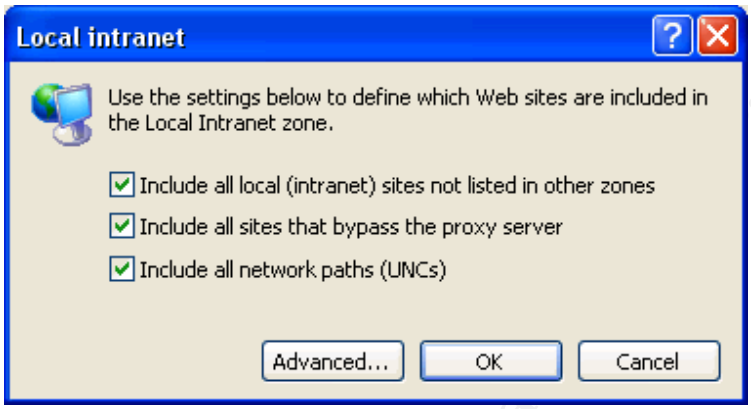


Figure 3 Local Intranet Zone Settings [4]

Trusted Sites and Restricted sites can be added manually by the user and have the option to authenticate the sites using Server verification using https.

For each of the 4 zones the security settings can be customised:

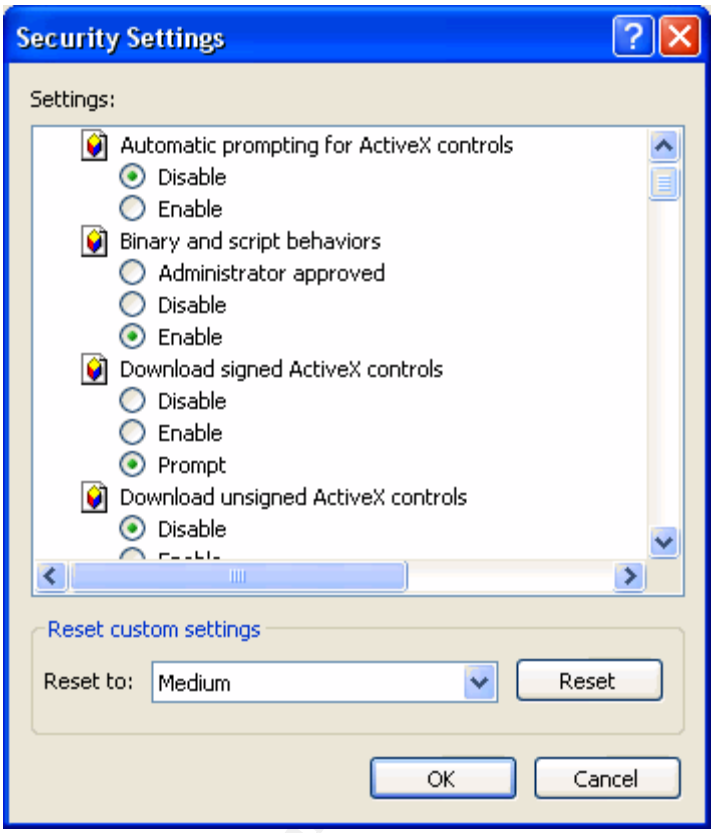


Figure 4 Security Settings for Zones

Typically the Intranet zone would have the least restrictive settings, such as Active Scripting switched on, ActiveX controls switched on. All these zones are customisable.

Sans recommends the following settings: [2]

Under Scripting, select Disable for "Allow paste operations via script" to prevent content from being exposed from your clipboard.

Note: Disabling Active Scripting may cause some web sites not to work properly.

ActiveX Controls are not as popular but are potentially more dangerous as they allow greater access to the system.

Select Disable for "Download signed ActiveX Controls".

Select Disable for "Download unsigned ActiveX Controls".

Also select Disable for "Initialize and script ActiveX Controls not marked as safe".

Java applets typically have more capabilities than scripts and it is good practice

to ensure that system operations and maintenance are undertaken from an account that does not have Administrator privileges.

Under Microsoft VM, select "High safety for Java permissions" in order to properly sandbox the Java applet and prevent privileged access to your system.

Under Miscellaneous select "Disable for Access to data sources across domains" to help avoid simple Cross-site scripting attacks.

Please also ensure that no un-trusted sites are in the Trusted sites or Local intranet zones as these zones have weaker security settings than the other zones