White Paper

McAfee

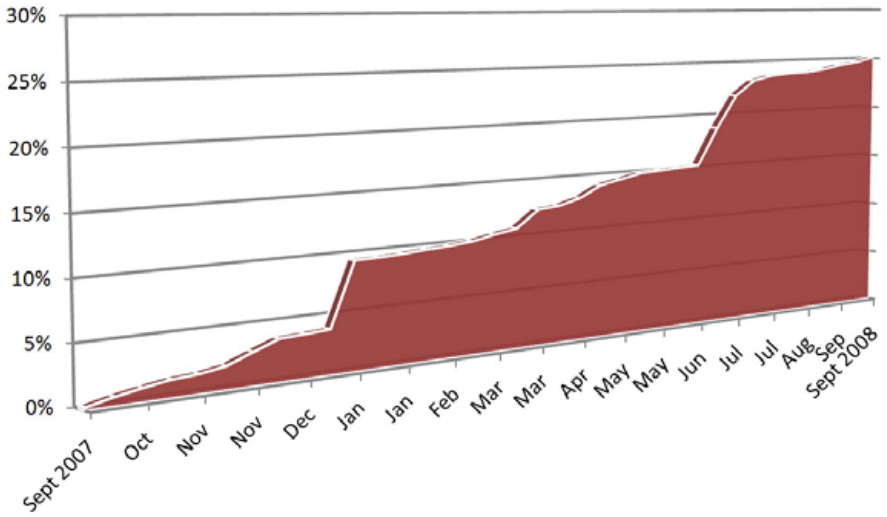
Web Browsers: An Emerging Platform Under Attack

By Christoph Alme

Table of Contents

Web Browsers: An Emerging Platform Under Attack

The widespread use of highly interactive "rich client" web applications for e-commerce, business networking, and online collaboration has finally catapulted web browsers from straightforward HTML viewers to a full-blown software platform. And as corporate users are performing a significant portion of their work on the web, whether it's researching or collaborating, the safety of the underlying platform is critical to the company's success.



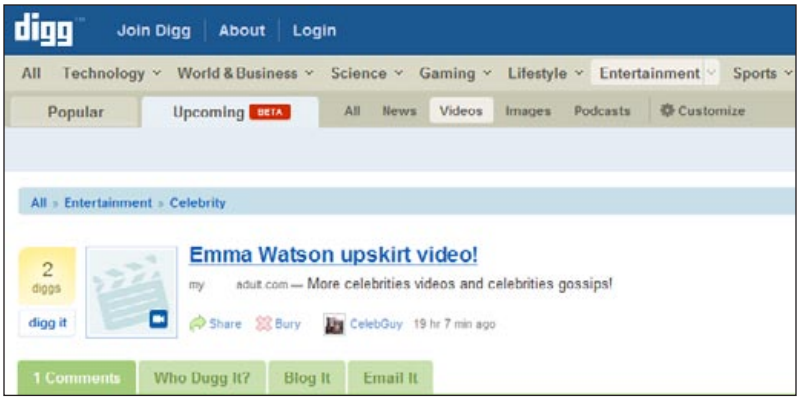
Growth of exploits on malicious or compromised websites, by unique variants in the wild, 2007-2008.

As use of the web and browser capabilities grows, so does the prevalence of web-borne malware and compromised websites. The "classical" attack vector of using malicious executable attachments in emails has been overtaken by the following attack vectors:

- Spammed emails now contain only a link to a malicious site
- "Web 2.0" sites—whether weblogs, social networking or portal sites—are spammed with links to malicious sites
- Legitimate sites are compromised and misused to either host malicious code or link to a malicious website
- Malicious video banners are placed into advertisement networks and once displayed on legitimate websites, they refer the unsuspecting user to a malicious site
- Popular search terms are used to advertise and drive (search query) traffic to a malicious website. In a recent case[1] in Germany, attackers used Google AdWords to attract users who searched for "flash player" to the attacker's fake Adobe-look-alike site

Add to that the web's classical social-engineering trick of the "missing video codec," where visitors to fake websites, promising adult content, are tricked into believing a video codec needs to be installed in order to play back the advertised videos.

1 http://www.pcwelt.de/start/sicherheit/firewall/news/185059/vorgeblicher_flash_player_10_ist_adware/



Link to malicious website placed on popular Web 2.0 site digg.com.

Eventually, all these vectors lead to a malicious website, whether an intended or compromised one. These sites either use exploits in an effort to install malware on the visitor's PC silently ("drive-by infection"), or trick the visitor into downloading a malware executable through some social-engineering technique ("drive-by download").

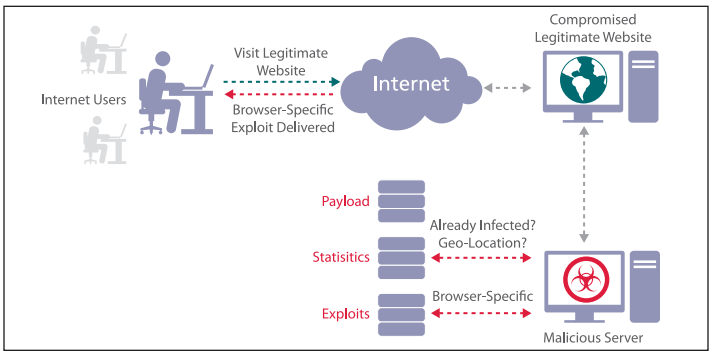
Attackers continue to use older exploits, such as for Animated Cursors (MS07-017) and Microsoft Data Access Components (MS06-014) vulnerabilities, but they also employ more corporate-specific exploits—such as for a buffer overflow vulnerability in a WebEx ActiveX control[2] (CVE-2008-3558)—from time to time.

At the same time, web browsers are the critical platform for both current and future business applications, and are the target of sophisticated web-borne malware attacks. The next section will introduce today's web attack reality, followed by an in-depth look into browser security and its current strengths and limits. The last section, "Plug-in Update Haze" gives an overview of the update methodologies of popular browser plug-ins today.

The Contemporary Attack Backbone

The malicious websites, to which these attack vectors inevitably lead, are usually set up with the help of an "exploit toolkit." The first widespread, infamous toolkit was "web Attacker," later known as "MPack."

Exploit toolkits are typically a collection of PHP scripts connected to a back-end database server. They come with a web administration interface that allows the attacker to remotely command and control a campaign.



The common workflow of web-borne malware attacks.

2 https://forums.symantec.com/syment/blog/article?blog.id=vulnerabilities_exploits&message.id=167#M167

These toolkits are sold with different sets of preinstalled exploits. Once set up, an attacker needs to direct traffic to his malicious website, for example, by injecting script references or invisible IFRAMEs into compromised sites. In many cases this happens by exploiting SQL-injection vulnerabilities in insecure web applications, or by utilizing stolen FTP credentials.

CVE	Title
CVE-2008-2463	Microsoft Access Snapshot Viewer Arbitrary File Download Vulnerability
CVE-2008-1309	RealPlayer ActiveX Control "Console" Property Memory Corruption
CVE-2007-5659	Adobe Reader and Acrobat JavaScript methods buffer overflow vulnerabilities
CVE-2007-5327	CA BrightStor ARCserve Backup Multiple Vulnerabilities
CVE-2007-0018	NCTsoft NCTAudioFile2 ActiveX Control Remote Buffer Overflow Vulnerability
CVE-2007-0015	Apple QuickTime RTSP URL Handling Buffer Overflow Vulnerability
CVE-2006-5820	AOL SuperBuddy ActiveX Control "LinkSBIcons()" Vulnerability
CVE-2006-5745	Microsoft XML Core Services Vulnerability
CVE-2006-4777	DirectAnimation ActiveX Controls Memory Corruption Vulnerability
CVE-2006-3730	Windows Shell Remote Code Execution Vulnerability (webViewFolderIcon)
CVE-2006-0003	Microsoft Windows MDAC Vulnerability
CVE-2005-2127	COM Object Instantiation Memory Corruption Vulnerability (Msdss.dll)

Table 1: Extract of ready-to-use exploits available with the toolkit "El Fiesta."

When users visit a compromised, previously legitimate, website, the site silently refers them to the malicious server. Here, their browser and plug-in versions are determined to deliver only exploits with the highest chance of "success." Also, the geographical location of the victim is sometimes determined to limit campaigns only to certain regions.

Advanced browser stats			
Browser	Visits	Exploited	Percent
MSIE v6.0	291	134	46.05%
MSIE v7.0	243	54	22.22%
Firefox v2.0.0	177	32	18.08%
Firefox v3.0.1	53	10	18.87%
Opera v9.51	18	6	33.33%
Opera v9.27	8	3	37.5%
Opera v9.25	6	3	50%
Opera v9.50	9	2	22.22%

Screen capture of infection statistics of an active "Sploit 2.5" toolkit installation.

Exploits are delivered only once to each visitor, to make it harder for security researchers to analyze the malicious server. Upon successfully exploiting the vulnerability, the attacker's payload is downloaded and installed on the victim's computer. This is usually some kind of a password-stealing spyware, looking for online banking, web mail, ICQ, FTP, or Windows logon credentials. Recently, the downloads are often performed in covert channels, that is, the executable is hidden, for example, inside an innocent-looking graphics file.

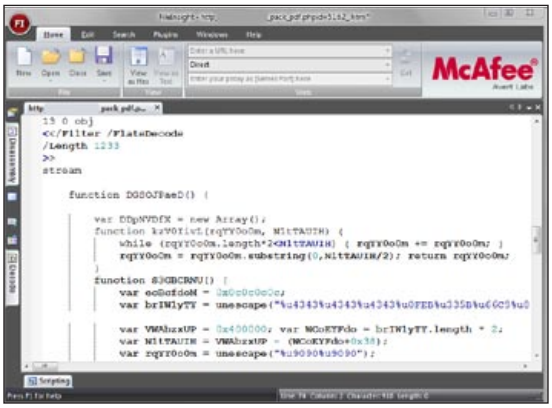
"El Fiesta" toolkit web statistics interface.

The toolkit keeps track of which exploits have been successful, how often users of which browsers have been exploited, geographical location, and so on.

Due to the widespread availability of Portable Document Format–reading software across all browsers, exploits for Adobe Reader vulnerabilities were added to major exploit toolkits in 2008. PDF is the document exchange format most commonly used in corporate environments. One toolkit, the "PDF Xploit Pack," targets PDF exclusively.[3]

Likewise, the toolkit "El Fiesta" extended its PDF exploit capabilities lately; it now dynamically generates unique PDF documents on the server side.

3 http://www.trustedsource.org/blog/153/Rise-Of-The-PDF-Exploits

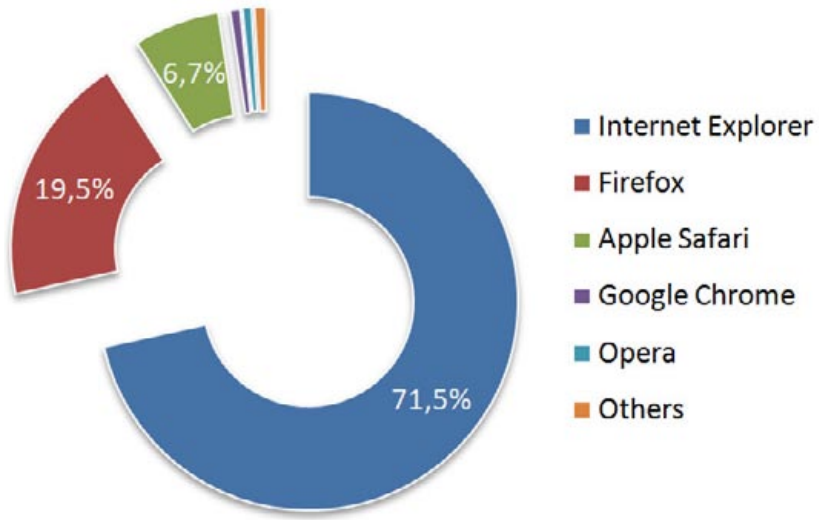


Malicious PDF opened for analysis in FileInsight.[4]

Each visitor gets served a PDF exploit with randomly chosen parts that disable identification by checksum or signature of the affected script code parts.

Browser Core Security

Web browsers are the critical platform for current and future business applications, and are equally the target of sophisticated web-borne malware attacks, as described in the previous section. This section outlines security aspects of the browser core, before delving into the abyss of browser plug-ins.



Browser prevalence by September 2008. Data courtesy of Net Applications.[5]

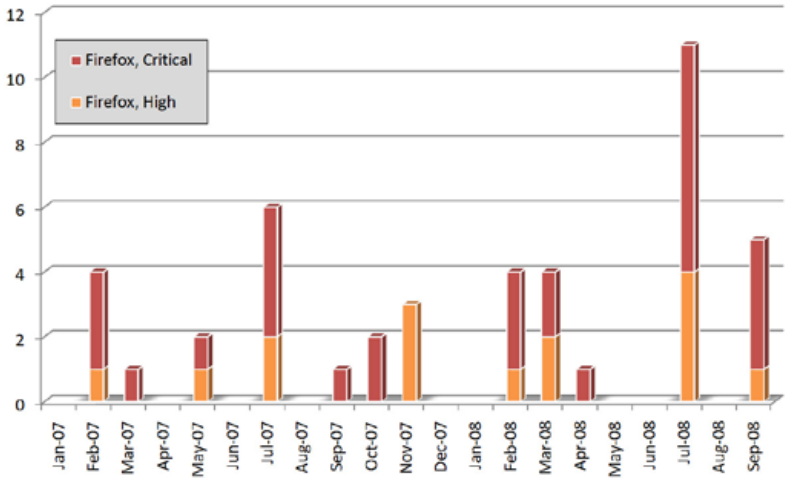
According to the latest statistics, Microsoft's Internet Explorer remains the most prevalent browser, especially in corporate environments. Nevertheless, Firefox has significantly closed the gap. Third is Apple's Safari browser, which can be attributed to both the popularity of iTunes—which uses Safari on Windows—and to the growing popularity of the Mac OS X platform as well.

4 http://www.webwasher.de/download/fileinsight/
5 http://marketshare.hitslink.com/report.aspx?qprid=0

Germany's primary television channel, ARD, reporting a warning from the Federal Office for Information Security not to use Google Chrome for everyday usage in its current state.

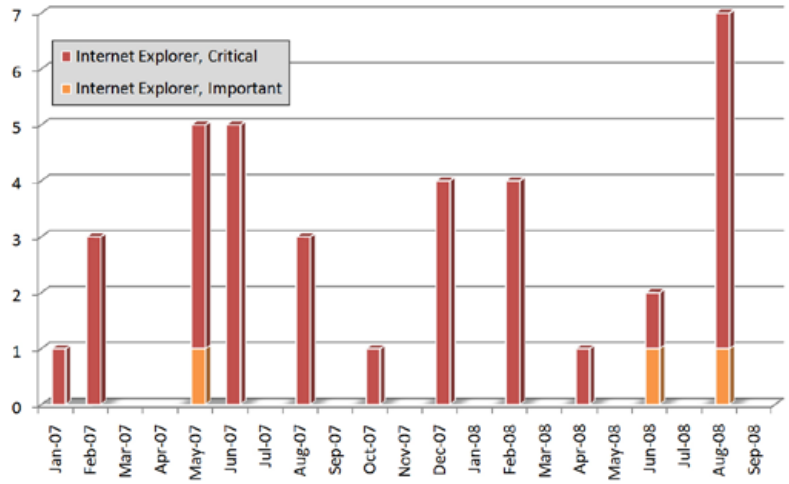
Although Google correctly tagged its new Chrome browser as a beta, this did not stop it from capturing the number-four spot among browsers and from overtaking Opera's position within only a few days after Chrome's launch. Malware authors did not overlook Chrome's debut; it was hit by a buffer overflow vulnerability in its early days of existence. Such parsing zero-day vulnerabilities, which can be misused for arbitrary code execution through nothing more than malformed HTML, can quickly lead to widespread exploitations in the wild, as was the case with Internet Explorer's "Windows URI Handling Vulnerability" (MS07-061).

Using an outdated version of the open-source WebKit as the underlying HTML-rendering engine, Chrome also reintroduced the "Carpet Bomb" vulnerability, which hit Safari users early in 2008. Using the outdated engine makes it especially dangerous for users of Internet Explorer to test-drive the new Google browser, as the vulnerability is greatest in exactly this scenario: Using Chrome temporarily, a user would visit a malicious site that exploits the vulnerability to drop an arbitrary file on the user's desktop. Not noticing this silent drop, the user then returns to Internet Explorer, which loads and executes the arbitrary file.



Vulnerabilities affecting Firefox browser core, 2007-2008.

Looking at the number of important and critical vulnerabilities fixed in the major browsers in 2007 and 2008, no decrease in vulnerabilities and the associated malware problems is evident so far.



Vulnerabilities affecting Internet Explorer browser core, 2007-2008.

Protection from Code Execution Exploits

The remote execution of arbitrary code remains the most dangerous threat to web browsing today. With an unpatched code execution vulnerability in your browser, visiting a potentially compromised site can silently run an attacker's code in the context of the logged-in user and install any kind of malicious payload, such as password-stealing malware, in a "drive-by infection" attack. This requires no user interaction at all.

Remote code execution is in almost all cases done by exploiting a buffer overflow vulnerability. Such a vulnerability occurs when a program expects data supplied from the Internet to be in a certain format, yet without actually verifying this. By supplying crafted malformed data, an attacker can trigger the vulnerability and alter the path of code execution in the client software to execute machine code (so-called shellcode) that was supplied by the attacker along with the malformed data.

Browswer	/GS Stack Buffer Overflow Protection	/NXCOMPAT Heap Buffer Overflow Protection
Internet Explorer 7.0	Yes	No
Firefox 3.0	Yes	Yes
Apple Safari 3.1	Yes	No
Google Chrome 0.2	Yes	No
Opera 9.5	No	No

Table 2: Security-relevant compiler protections used in the most popular browsers (on Windows).

The latest compiler versions have introduced generic protection measures against both stack-based and heap-based buffer overflows. With such protection compiled into the browser, the browser would be terminated by a buffer overflow before the malicious code gets executed. The table above shows that at least the protection from stack-based buffer overflows is enabled in most browsers today, while protection from heap-based buffer overflows is currently available only in Firefox.

No Phishing Grounds

Code execution exploits are a purely technical attack. The best defense for all users is to keep up to date with patches and anti-malware updates. Administrators can also employ both a gateway anti-malware system with buffer-overflow and shellcode-detection heuristics as well as a host intrusion prevention system (HIPS), which provides generic buffer-overflow protection for common buffer-overflow exploits and specific browser vulnerabilities.

Today's major browsers, Internet Explorer and Firefox, both ship with built-in phishing protection through a list of known phishing sites. For example, using Firefox on Mac OS X would block access to known phishing sites, giving users additional protection over the platform's default browser, Safari.

Firefox running on Mac OS X, blocking access to a known phishing site.

IT administrators need to decide whether they can enforce a common browser policy across all platforms and devices, or if they are better served using central phishing protection at the corporate gateway.

Relief from Cross-Site Scripting

Cross-Site Scripting (XSS) refers to a type of vulnerability that erroneously allows script code from a hostile website to be executed in the context of another (trusted) site, thereby allowing the hostile site to steal data (such as logon credentials) associated with the trusted website. This can happen, for example, by passing the malicious script code along with a search query or in a web form being posted to the vulnerable web application. The vulnerable web application then fails to "normalize" the script code and instead returns content that includes the attacker-supplied code. The browser will next render the content and also execute the injected script code. At this point the browser "believes" the code belongs to the trusted website's context and will thus get full access to the site's data.

The "Same Origin Policy" normally prevents such access in today's web browsers; that is, only script code hosted on site X is allowed to access data of site X. The simplest form of an XSS vulnerability is the remote injection of script code into the response of a web server; such attacks can usually be determined by the presence of "<script>" tags in the request URL (in its various encoding forms, such as "%3Cscript" and so on), or by the presence of data-access script functions such as "document.cookie."

The upcoming Internet Explorer Version 8, currently in beta, introduces a quite promising protection layer[6] against this type of cross-site scripting. Upon receipt of the web server's response, the browser will compare the request URL and the response body. It looks for matching fragments that indicate the presence of script tokens supplied in the URL parameters and recurring in the body.

Type	Affected Software, Website, or Device
XSS	Mambo CMS http://www.securityfocus.com/archive/1/487128/30/0/threaded
XSS	Drupal CMS http://drupal.org/node/295053
XSS	Joomla CMS http://www.securityfocus.com/archive/1/485676/30/0/threaded
XSS	Apache Tomcat http://tomcat.apache.org/security-6.html
XSS	PayPal http://news.netcraft.com/archives/2008/05/16/paypal_xss_vulnerability_undermines_ev_ssl_security.html
Cross-Zone Privilege Escalation	Skype http://skype.com/security/skype-sb-2008-001.html
XSS	Facebook http://www.xssed.com/news/69/Facebook_vulnerable_to_XSS._Over_70_million_users_are_at_risk
CSRF	Linksys Router http://www.heise-online.co.uk/security/Crafted-web-site-switches-off-router-firewall--/news/101650

Table 3: Extract of Cross-Site Scripting (XSS) and related vulnerabilities in 2008.

Not all variants of XSS can be fended off so simply. One clever variant of XSS is called Cross-Site Request Forgery (CSRF). In 2008, one CSRF vulnerability was exploited in the wild in the Linksys router, disabling the router's firewall. Such CSRF attacks do not involve any script code, but rather exploit vulnerabilities in the target web application merely by tricking a user into opening a website that in turn invokes a URL with application-specific parameters. For example, imagine your router's web interface allows settings to be edited by invoking a specific URL. The following example is taken from the Linksys CSRF attack:

```
https://192.168.1.1/apply.cgi?submit_button=Firewall&change_action
=&action=Apply&...
```

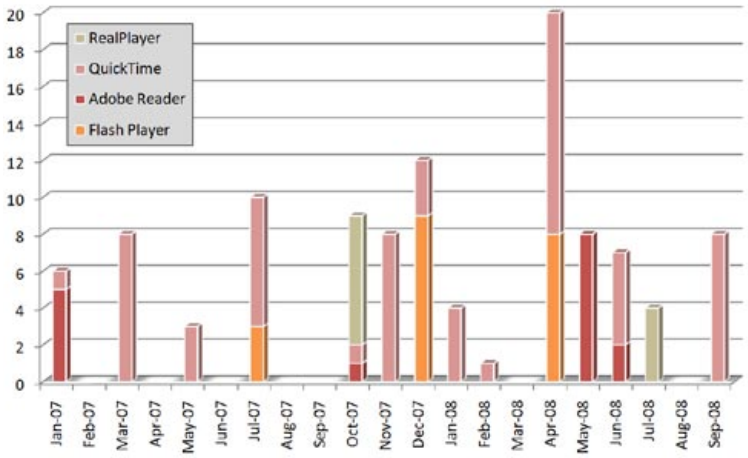
An attacker would put such a URL onto the hostile website, for example, as the source of an "<img>" tag; once the browser renders that page, it will implicitly invoke this URL. Fortunately, such attacks are possible only when the user is already logged into the application (such as a router administration interface) that is under attack.

Browser Plug-ins Security

Although most malware threats come via the web, the browser core itself is rarely the target of exploitation. Rather, any sort of plug-in—whether for document reading, interactive content, or ActiveX controls—is subject to attacks in the wild.

So no matter how resistant the Internet Explorer browser has become in recent years, a buffer overflow such as "Vulnerability in Cisco WebEx Meeting Manager ActiveX Control" (CVE-2008-3558), which was exploited in the wild just two weeks after the vulnerability was known, would still be a major concern for any corporate IT administrator who manages WebEx. The vulnerable control would be updated only upon connection to a patched WebEx server.

6 http://blogs.msdn.com/ie/archive/2008/07/02/ie8-security-part-iv-the-xss-filter.aspx



Vulnerabilities affecting popular browser plug-ins, 2007-2008.

Browser plug-ins are usually installed on demand; as soon as you visit a site and want to view, for example, some interactive content (Flash movie, Java applet, etc.), you may be asked by your browser to install the missing plug-in. Be leery of malicious websites trying to exploit this as a social-engineering trick; one common ruse is to trick visitors into downloading a "missing video codec."

Over time, most users enhance their browsers with more and more plug-ins that are needed to properly display various websites. Add to that the side-by-side deployment of some plug-ins in multiple versions—for example, the Java and .NET runtimes—and it is easy to see that the concerns about browser plug-in security are not going away soon.

Browser Plug-in	Installed at % of Users	Supports
Flash Player	98.8	All
Adobe Reader	> 80	All
Sun Java	84	All
Windows Media Player	82.2	Internet Explorer
QuickTime Player	66.8	All
Real Player	47.1	All

Table 4: Data taken from ETH Zurich Tech Report No. 288, "Understanding the web browser threat."[7]

7 http://www.techzoom.net/publications/insecurity-iceburg/

Exploit Protection and Its Limits

Similar to the situation with web browsers, protection from stack-based buffer overflows is already built into most browser plug-ins. Heap-based buffer overflow protection, on the other hand, is still lacking.

Browser Plug-in	/GS Buffer Overflow Protection	/NXCOMPAT Heap Buffer Overflow Protection
Flash Player 9.0	No	No
Adobe Reader 9	Yes	Yes
Sun Java 6	Yes	No
QuickTime 7.55	Yes	Yes
Real Player	Yes	No
Silverlight	Yes	Yes

Table 5: Security-relevant compiler protections used in the most popular Windows browser plug-ins

Using an unsafe plug-in renders the browser unsafe as well. For example, the Java and .NET runtimes can be misused to allocate memory with executable privileges turned on, allowing the malware to more easily work around data execution prevention (DEP), as shown by Sotirov and Dowd in their BlackHat 2008 paper "Bypassing Browser Memory Protections."[8]

New JavaScript Hideouts

Because anti-malware scanners can inspect JavaScript on web pages, attackers are obfuscating their malicious code to avoid easy detection. Generic detection and unpacking methods did not put an end to this shell game, as attackers are now moving their malicious JavaScript code into the file formats handled by browser plug-ins. Flash videos and PDF documents, for example, are now used to carry malicious JavaScript code.

A Ghost in Your Clipboard

As noted in the introduction, today's attack vectors include the spamming of weblogs and forums with entries and links to malicious websites. These do not necessarily need to be explicitly created by the attackers: In a recent case, innocent users' browsing sessions were hijacked to indirectly perform that task.[9]

The attackers found a method buried in the scripting capabilities of Flash that allowed the placement of text onto a user's clipboard. Equipped with this seemingly simple functionality, the attackers then created a Flash video that would overwrite the clipboard with fraudulent text content and a malicious link every couple of seconds, and they fed this Flash video as a banner ad into advertisement networks. Once the banner appears as an advertisement on some legitimate site that users visit, the fraudulent message is silently placed onto the clipboard. Users may have thought they copied some interesting text passage onto the clipboard and, meanwhile, they switch to a favorite blog or forum and try to post that interesting text. Instead, they will accidentally post whatever was placed last onto the clipboard, which in this case is the attacker's text and link. Thus the unaware users do the distribution work for the attackers. The only solution to stop the "ghost" in your clipboard is to shut down the browser that runs the malicious Flash video.

This technique has been used in the wild to promote links to so-called rogue anti-spyware products, that is, fake products that try to scare[10] users by presenting them with a fake system scan and claiming that the computer was infected by malware.

8 http://taossa.com/archive/bh08sotirovdowd.pdf
9 http://www.trustedsource.org/blog/145/Rouge-Flash-ads-hijack-your-clipboard
10 http://www.trustedsource.org/blog/148/Fake-Madonna-video-turns-the-blue-screen-on

Plug-in Update Haze

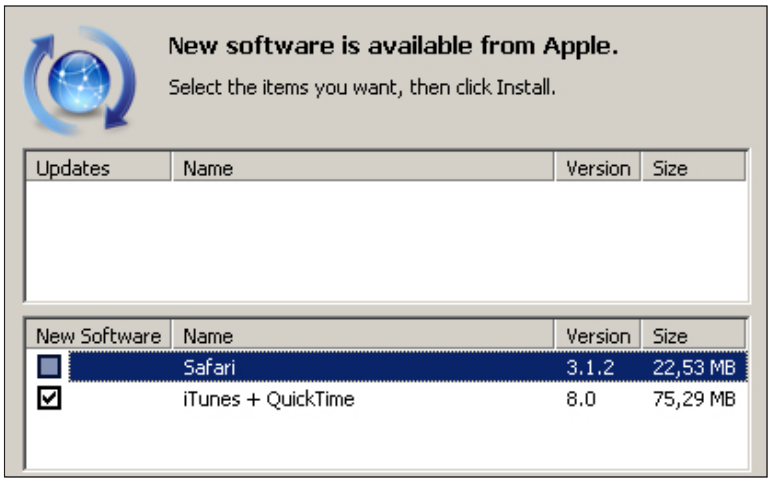
Corporate IT administrators need to keep an eye on deployed software and how updates are rolled out, but they rarely have all the required budget, tools, and personnel at hand. In today's enterprise environments, updates for Windows and other Microsoft applications are commonly rolled out through in-house "Windows Server Update Services."

Browser Plug-in	Interval	Download	Installation
Flash Player	Monthly	Notify	Notify
Adobe Reader	Weekly	Automatic	Notify
Sun Java	Monthly	Notify	Notify
Windows Media Player	Windows Update	Automatic	Automatic
QuickTime Player	Weekly	Notify	Notify
Real Player	Twice a week	Automatic	Automatic
Silverlight	Windows Update	Automatic	Automatic

Browser plug-ins and their update check intervals.

Third-party plug-ins typically are not distributed through Microsoft's Update Server. In a case in which vulnerable versions of Adobe's Flash Player were included with Windows XP by default, Microsoft issued Security Bulletin MS06-020 to patch them. Third-party plug-ins come with their own updaters and may be missing control features needed in large enterprises. Plug-ins that notify users about an update rather than installing it are a problem in a corporate environment, simply because they leave the update decision to users, who are annoyed and distracted from their work.

Furthermore, desktop computers may not be allowed to download executable patches directly from a vendor's website, due to strict security policies. Most important, users should not be doing their normal day-to-day work with an administrator account but with a normal restricted user account instead. Thus applying patches is often not possible for them.



Updater for QuickTime, advertising the Safari browser and iTunes, even though the installed QuickTime version was already up to date.

Another aspect of third-party plug-in update methodologies is that they sometimes advertise applications in addition to the one users want. A number of vendors do this, including Adobe, Apple, Google, Mozilla, and Yahoo. Apple's QuickTime, for example, ships with Apple Updater, which offers to install Safari and other applications. This is a good way for Apple to distribute its web browser, but it puts another burden on administrators, who have to manage another application—one that might not be sanctioned.

Conclusion

The first layer of malware defense should be placed at the network perimeter and be managed centrally. This helps the IT person in charge sleep better. This layer alone can't replace proper patch management and rollouts, but it keeps desktop computers with outdated or misconfigured local security software protected while they surf the Internet, and it can identify and isolate desktop computers that are infected—preventing the leakage of sensitive data and credentials. And because scarce false-positives are less problematic at the gateway level, more aggressive heuristics are possible that permit the highest proactive detection rates.

About the Author

Christoph Alme serves as head of Anti-Malware R&D at McAfee's Network Security business unit, formerly Secure Computing. He is responsible for overseeing anti-malware research and development. Alme is the inventor of several patent-pending key technologies in the field of proactive malware detection. Before joining Secure Computing, he worked at SAP and BMW.

McAfee®

McAfee, Inc.
3965 Freedom Circle
Santa Clara, CA 95054
888 847 8766
www.mcafee.com