Malicious Firefox Extensions

Philippe Beaucamps, Daniel Reynaud
philippe.beaucamps[at]loria{dot}fr, reynaudd(at)loria_dot_fr
Loria - Nancy, France

June 2008

The purpose of this paper is to warn Mozilla Firefox users about the risk associated with browser extensions, especially the fact that Firefox extensions are more dangerous than ActiveX controls in Internet Explorer. Interestingly, the lack of security of ActiveX controls is well known whereas the security of Firefox extensions seems to be ignored. This danger is platform-independant, Microsoft Windows as well as Mac OS and Linux are vulnerable as long as Firefox is used. Finally, we only consider Firefox but virtually every Mozilla application implementing an extension mechanism is subject to this threat, such as Thunderbird.

1 Introduction

Extensions are the equivalent of ActiveX controls and Browser Helper Objects in Internet Explorer: they allow to interact at different levels with the browser and to control and modify its behavior. They come as:

- Classic extensions

- Language Packs

- Themes

They have the following characteristics:

- They can be coded in scripting languages (Javascript), in high-level languages (Java, Python) and low-level languages (C/C++).

- They can contain native libraries (DLL on Windows) and call their functions, but they can also contain native programs (such as EXE files on Windows) and launch them.

- They are cross-platform (but the use of native libraries or code can make them platform dependant).

An extension can be installed in different ways:

- From the browser, using the addon manager or from webpages;

- By an external program (a dropper), that will install it without the user's consent.

This infection mechanisms are similar to the situation with ActiveX controls in MSIE.

2 Stealth Mechanisms

An extension can hide its presence using several means:

2.1 Addon Manager

An extension can disappear from the addon manager using several techniques:

- if it is installed in the system's extensions directory, there is a `hidden` property to activate in the extension's installation manifest `install.rdf`;

- in the user profile directory, the file `extensions.rdf` is an XML document listing the visible extensions. An extension can thus edit its entry in `extensions.rdf`;

- The extension can modify the code of the addon manager (*overlay*), and hide its presence by modifying the CSS of the manager;

- ...

2.2 Thwarting Detection

All the usual polymorphism and packing techniques apply, as with traditional malware.

2.3 Viral Behavior

An extension can replicate and infect other extensions (either global or profile-specific). Therefore, if you delete what you identified as a malware extension, it might continue running if it infected other already installed extensions. Moreover, signed extensions (such as the Google or Yahoo! Toolbar) can also be infected since their integrity is only checked during installation.

2.4 Installation and Removal

As stated earlier, extensions can be installed by dropper programs. This operation is totally silent in Firefox 2. In Firefox 3 there is a warning box saying "a new component has been installed" with no more details, we are currently working on a way to bypass this warning.

2.5 Malicious Activity

One of the main challenges of malware authors is to bypass personal firewalls and antivirus products. The usual technique on Windows system was to inject the malware as an MSIE thread, so that the malware activity looks like it comes from the browser. This is possible with Firefox extensions without having to write native code, any interesting payload can be programmed using Javascript APIs in Mozilla products. It also becomes useless to perform "noisy" operations such as registering as a service or adding a registry key to survive a reboot, since it is a fair assumption that the browser is permanently running.

3 Application Programming Interfaces

Firefox exposes an API called XPCOM (Cross Platform Component Object Model). This API is very powerful, it is cross-platform and can be called directly from Javascript code. The Firefox source code contains many examples with this API, for instance there is a documented way to create server sockets directly from Javascript code.

These functionalities allow to interact with Firefox and with the underlying system. They allow:

- read/write access to the *Document Object Model* (DOM) of the currently opened documents. This means the DOM can be modified before being displayed to the user, and before content is submitted to the server. Therefore, it is possible so spy on forms in order to steal passwords and alter user submissions. These techniques are currently aggressively used in banking malware for MSIE.

- *cleartext* access to stored *passwords*.

- read/write access to the *filesystem*.

- access to the *network* through client and server sockets. This allows traditional botnet and spam relay behavior.

- access to *native code and libraries*, granting the same rights on the system as Firefox

4 Conclusion

To sum things up, Firefox extensions are dangerous for the following reasons:

- **Simplicity of Javascript**, it becomes straightforward to write powerful malware with little or no programming experience;

- **Web Applications**: the browser is increasingly used for the manipulation of sensitive data: bank account management, e-mail, shopping, document sharing and so on;

- **Browser Ubiquity**: it is the consequence of the point above, virtually every system needs a browser and it is likely to be running permanently;

- **Platform Independance**: a malicious extension developed for Firefox will work on every platform supported by Firefox;

- **No Security Policy**: once an extension is installed, it runs at full trust

By combining the above functionalities, the result is an advanced malware development platform, documented, cross-platform and easy to use.

Note however that it is not the result of a vulnerability in Firefox but the consequence of some extensions management flaws and the absence of a security policy. This is not exactly new either, the problem and workarounds have already been detailed in 2007 (Louw, Lim, Venkatakrishnan - Enhancing web browser security against malware extensions), but the results have mostly been ignored, and might remain so until a security disaster happens, such as a large scale malware for Firefox with serious consequences.