Stupid JavaScript Security Tricks

Walter Cooke, CISSP

W. J. Cooke & Associates Ltd.
3216 Albert Street
Halifax, NS B3K 3M9
Canada
Phone (902) 453-4995; Fax (902) 453-4996
cooke@uncle.com
http://fox.nstn.ca/~cooke/

Abstract: Many offices now provide "Internet access to the desktop" as just another tool for staff to use in their everyday work. However, just because computer users sit behind a network firewall, this does not mean that their World Wide Web browsing systems are secure. Web browsers such as Netscape Navigator and Microsoft Internet Explorer have been found lacking when it comes to enforcing privacy and security (confidentiality, integrity, availability). Some of these issues in the use of the JavaScript language are explored, and some example scripts that exploit these issues are shown.

Browsing The Web

Browsing the World Wide Web (WWW) has become a hobby, work tool, and even a full time profession for many. However, newer versions of web browsers such as Netscape Navigator (NN) and Microsoft Internet Explorer (IE) now have programming languages that allow a web server to exploit both the programmable features of, and the vulnerabilities in the User's browser software, to the web server's advantage.

Netscape Navigator holds about 70% of the browser market share, and its "JavaScript" scripting language provides a good example of how a number of nasty security "tricks" can be played on the innocent web user. This does not mean that other browsers are safe to use – each one has its own set of security problems, some identical to the ones found in JavaScript-enabled browsers. At the same time, there seems to be some confusion between "Java" the SUN Microsystems-invented client/server programming language, the "JavaScript" scripting language from Netscape Communications, and the "ActiveX" scripting language from Microsoft. Each one provides programming functionality to the server/browser web interaction. But each one is a different and distinct product, and they should not be thought of as providing identical functionality, or that each comes with the same security concerns. Java is by far the best programming environment with the safest security model, while ActiveX provides the least secure programming functionality inherent in the major browser software packages.

The JavaScript Security Tricks

As shown on the accompanying slides, there are a number of potential security problems associated with JavaScript, most of which can be solved by keeping up with the most recent version of your browser. However, some of the security problems are inherent in any application which has scripting or programming capabilities. Also, an informal experiment set up at the author's web site http://fox.nstn.ca/~cooke/ has found that while the majority of Users do browse the web site with a version three (or greater) browser, a significant minority of Users are still using a version two browser that allows the web server to capture the User's eMAIL address and name. Do Users regularly upgrade their web browsers? The answer is no – large numbers do not do so, even though both NN and IE upgrades are available for free from many sources.

The JavaScript security tricks can be divided into three areas: Confidentiality, Integrity, and Availability problems. Confidentiality problems include capturing User information using passive means or by actively reading user files. Integrity problems come from JavaScript's ability to destroy, alter, or transmit the User's data. JavaScript can also be used to kill a User's browser session by opening an infinite number of windows, or simply crash the Operating System, which represents an Availability problem for the User. The provided script examples and web site hyperlinks demonstrate the wide range of security problems that JavaScript can play a role in creating or exacerbate problems in existing system limitations.

Finally, the use of JavaScript for totally inappropriate things such as cryptographic functionality are beginning to appear. Using a Common Gateway Interface language such as Perl or C would be a better way to provide this type of functionality.

A Ride on the Space Shuttle "JavaScript"

Currently, it remains a choice of security risk vs. enhanced browsing features, when deciding whether to switch browser programming capabilities on or off. If your workstation files are well backed up, you are not attached to a LAN, and you don't keep sensitive files on your hard disk, then you are probably relatively safe exploring the new hyperspace frontier. On the other hand, blindly accepting a high level of security risk on the web is as shaky as a ride on the Canadian space shuttle (made of birch bark and pine gum). A growing number of companies want to deploy Internet-based, mission-critical systems to their staff and customers. The legal and financial consequences of using high risk IT solutions may not be as dramatic as space shuttle accidents. Never the less, a prudent approach would be to include additional security enhancements to reduce the risk of being attacked by the next Internet security problem we discover.

More recent WWW security alerts do not directly relate to JavaScript. One such problem is "web spoofing," where a bogus web server is inserted between the victim's computer/web browser and the servers of genuine web sites. Another involves the use of bogus DNS entities. However, this does not mean that all of the loop holes in NN or IE have been plugged. Users should remain vigilant and upgrade to the most recent version of their preferred browser on a regular basis. And in the meantime, only enable ActiveX, Java, and JavaScript features *after* connecting to known Internet and Intranet servers that you trust. Keep these features switched OFF by default.

The following pages contain the slides used for this presentation at the conference.

Caution! Kids: don’t try these tricks at home! We are trained professionals!