



**IntelliSIGHT™**

Security beyond the edge

# Google's New Chrome Browser: A Cursory Security Review

Sept. 19, 2008

## Contents

Overview .....	3
Architecture .....	3
Known Vulnerability Considerations.....	4
iSIGHT Partners' Original Vulnerability Discovery.....	4
Conclusions .....	4

## Overview

The hue and cry of "software as a service" has not abated, and the Chrome Browser is an entry that positions Google to benefit from such trend. While still in Beta release, Google Chrome shows promise for its unique tab/process separation paradigm. iSIGHT Partners' Lab has taken a quick look at the Google Chrome Browser

## Architecture

The Google Chrome browser (Chrome) source distribution is comprised of nearly 30,000 files stored in nearly 1,400 directories. While many of these files are data files for testing the operations and functionality of specific browser modules, there are still more than 4,200 files in nearly 240 locations that contain source code. These files are grouped into several major categories:

About 15-20% of the files in the source distribution are labeled as "third-party" files that are used for building and testing Chrome. The third-party files include:

- *The lighttpd web server*
- *Cygwin, a Unix-like environment for Windows*
- *Python, a script-based programming language*
- *The zlib library, used for the 'deflate' method of compression*
- *The libpng library, used for handling PNG format images*

While this is a source distribution, many of these files used to test and build Chrome are included as binary files. By including the binaries to the tools, the time to compile Chrome is considerably reduced, since tool dependencies do not need to be verified.

Also distributed in binary form is the "Google Gears" plugin, which is a Rich Internet Application that allows for the creation and execution of Desktop web Applications.

Chrome uses the WebKit open-source browser engine for rendering HTML content, which is the same engine used by Apple's Safari browser and the Adobe AIR runtime environment. The version of WebKit used by the evaluated beta release of Chrome is 525.13, which is the same version used in Safari 3.1. The latest version of Safari, 3.1.2 as of Sept.14, 2008, is built using WebKit version 525.19.

A major design change in Chrome over other popular browsers is the separation of processes running under a browser tab. Unlike the monolithic structure of other browsers, each tab runs its own process in its own memory space, and closing a tab also closes the process and frees the allocated memory. This capability is designed to improve memory management and increase application stability. The Chrome browser itself runs as the parent process to the tabbed child processes, and each child process is "sandboxed" and can only communicate with the outside world through a limited API, which is used to send messages back and forth between the parent and child processes. The code that allows this functionality is located in the "sandbox/" subdirectory.

As the child processes have only limited access to the system resources, the child processes must request certain functionality from the parent process, and the parent process is the final arbiter of whether a child is provided with additional functionality or resources. The behavior is further enforced with the overwriting of a number of low-level functions, allowing enforcement of this behavior at the

application level. Because of this, a sandboxed process cannot directly access certain system-level resources.

One significant exception to the above is the network protocols, which run without the restriction of parent permission. This exception improves performance, but it still provides a significant attack surface; any vulnerability in the networking code will tend to be as exploitable or dangerous as it would be with any other browser.

## Known Vulnerability Considerations

There have already been a significant number of vulnerabilities reported in Chrome. The vulnerabilities are very similar to those reported in Apple Safari prior to the current version and have even been attributed in some cases to the older WebKit library. While iSIGHT Partners continues to examine these vulnerabilities, it is likely that this is true. However, other vulnerabilities reported for the Safari browser will probably not impact Chrome, since Chrome uses (by default compiler settings) Google's V8 JavaScript engine, which is an open-source engine. Because of this change, several previously reported vulnerabilities that affected the WebKit component in Safari do not exist in the V8 engine, although this should not be used to conclude that the V8 engine is any more or less secure than the default WebKit engine.

## iSIGHT Partners' Original Vulnerability Discovery

Meanwhile, iSIGHT Partners' Lab has also already discovered a vulnerability in Chrome in the handling of HTTP traffic with a "Transfer-encoding" type of "chunked." By supplying a negative length, it is possible to cause the contents of the heap to be shifted by a controlled amount. While the thread that causes the heap modification will crash and trigger code (which eventually leads to the process being terminated), other threads that use the same heap may access the heap before the application quits. By manipulating the times at which attack events occur and sending data to Chrome with a specific timing and in a specific order, it may be possible for a remote attacker to cause the application to access data in the heap that will allow for arbitrary code execution. While an attack of this nature is unlikely to be highly reliable, code execution could occur without interaction other than visiting a malicious website. Additionally, because of the way processes are handled in Chrome, it is also likely that exploitability would be higher, due to the state of the application being more predictable, even after a long period of use. iSIGHT Partners will continue to examine this issue and will report the full details in the near future.

## Conclusions

Google Chrome, even though it is just in Beta release, is likely to continue to garner significant attention for several reasons. First, it is a Google product, and it is well known that Google extends the beta stage of software in order to attempt to absolve it from liability. This Google Beta stage does not prevent market penetration in a significant way; Google Desktop Search was in Beta for years, but it was still adopted for use on a day-to-day basis by hundreds of thousands of users. Because the computing world continues its inexorable move to web-based technology and because Google has a reputation for excellence, we can expect to see early adoption of Chrome by many users who may not be aware of the security implications of doing so.