curl SOS Fund Audit Fix Log

Identified Vulnerabilities

CRL-01-001 Malicious server can inject cookies for other servers (Medium)

Fix: https://github.com/curl/curl/commit/cff89bc088b
Advisory: https://curl.haxx.se/docs/adv_20161102A.html

VERIFIED

CRL-01-002 ConnectionExists() compares passwords with strequal() (Medium)

Fix: https://github.com/curl/curl/commit/b3ee26c5df75d97
Advisory: https://curl.haxx.se/docs/adv_20161102B.html

VERIFIED (comparison)
NO FIX IMPLEMENTED (timing attack)

Daniel Stenberg (curl): The timing issue is left without action and deemed to be not likely to be exploitable to any serious extent to cause damage.

Mike Wege (Cure53): *I question the declassification of the timing attack. The proposed hashing solution to the brute force attack would be easy enough to implement. At least it should not be forgotten, and added to the todo list.*

Daniel Stenberg (curl): first, it is rare that applications even use URLs with user+passwords from different users, as most applications actually run as a single user who then has access to all URLs and users and passwords. But even if not, a failed password check means that curl would instead proceed to setup a new connection instead of re-using one, which thus would add **much** more time to the operation than just the missed password compare. I can't see how a user can time just the failed password check and use that. There are many more factors around setting up a new request that also will vary in time and probably to a larger extent than the fluctuation of the string compare function. Or I'm not understanding the attack.

CRL-01-005 OOB write via unchecked multiplication in base64_encode() (High)

Fix: https://github.com/curl/curl/commit/efd24d57426bd7
Advisory: https://curl.haxx.se/docs/adv_20161102C.html

VERIFIED

Mike Wege (Cure53): *I question whether the preprocessor conditional is necessary. The optimisation is minimal, and it adds a dependency. Curl will be used in the current IoT trend,*

where memory is scarce anyway, while on 64bit-platforms the small runtime gain is questionable.

Daniel (curl): We could possibly have made it use “max value of size_t instead of UINT_MAX but I don’t think that is as easily accessible so it would’ve required some other trickery instead and thus added other quirks instead.

CRL-01-007 Double-free in aprintf() via unsafe size_t multiplication (Medium)

Fix: https://github.com/curl/curl/commit/8732ec40db652c
Advisory: https://curl.haxx.se/docs/adv_20161102D.html

VERIFIED

CRL-01-009 Double-free in krb5 read_data() due to missing realloc() check (High)

Fix: https://github.com/curl/curl/commit/3d6460edeee21
Advisory: https://curl.haxx.se/docs/adv_20161102E.html

VERIFIED

CRL-01-011 FTPS TLS session reuse (Low)

Fix: https://github.com/curl/curl/commit/1671d84b38ac61f0852e8ff2915fef3346dc53f7

Daniel Stenberg (curl): We lowered the security level of this issue to “very low” and committed it as a normal bug fix where we disable cross-protocol session id reuse. The attack scenario is very complicated and requires that the attacker has a lot of power in both ends.

Mike Wege (Cure53): *Reclassification accepted.*

VERIFIED

CRL-01-013 Heap overflow via integer truncation (Medium)

Fix: https://github.com/curl/curl/commit/53e71e47d6b816
Advisory: https://curl.haxx.se/docs/adv_20161102H.html

VERIFIED

Mike Wege (Cure53): *I question the error-prone indentation-style used in this fix (if statement with no braces). In particular the comment between the alternate case and the actual statement is highly discouraged.*

CRL-01-014 Negative array index via integer overflow in unescape_word() (High)

This issue was fixed in the same commit as CRL-01-013.

VERIFIED

CRL-01-021 UAF via insufficient locking for shared cookies (High)

Fix: https://github.com/curl/curl/commit/c5be3d7267c7
Advisory: https://curl.haxx.se/docs/adv_20161102I.html

VERIFIED

Mike Wege (Cure53): *I suggest removal of the backward 'goto fail'. There is no logical flow in that direction and the actual code reuse optimisation is minimal.*

Miscellaneous Issues

CRL-01-003 Ambiguity in curl_easy_escape() argument (Low)

Daniel Stenberg (curl): Due to the nature of this note (not a security vulnerability), it was taken to public discussion on the curl-library mailing list. There was no particular uptake or attention put to the issue even though a suggestion was made on how the API could be modified to perhaps reduce future mistakes better. My personal view is also that this is not big problem area to people so no big concern is raised.
https://curl.haxx.se/mail/lib-2016-10/0028.html

NO FIX IMPLEMENTED

CRL-01-004 Metalink provides an oracle (Info)

Daniel Stenberg (curl): So this is a remark on metalink as a concept rather than our implementation of it and, while I agree, I can't think of anything immediate that we can do to mitigate this concern. So this is left without action.

REJECTION ACCEPTED

CRL-01-006 Potentially unsafe size_t multiplications (Medium)

Partial fix: https://github.com/curl/curl/commit/0649433da53c7165f839e2

Daniel Stenberg (curl): One step taken is discussing a "safe realloc" function to help us avoid problems with wrapped counters that cause free-twice problems:
https://curl.haxx.se/mail/lib-2016-11/0087.html

PARTIALLY VERIFIED

Mike Wege (Cure53): *While this fix addresses the double-free issue, the particular overflow problems are not addressed yet.*

Daniel Stenberg (curl): Looking at our history, the realloc() side of the possible overflows seems to be the most serious outcome and I've not detect much other serious potential problems and thus I've focused on that case.

CRL-01-008 %n is supported in format strings (Low)

https://github.com/curl/curl/commit/71588c9aef8112025c7525d20f57eb367a947344 (MAX_PARAMETER remark only)

Daniel Stenberg (curl): We will continue to discourage users from using the printf() functions we provide, but %n needs to remain supported until the day we remove those functions from the API.

REJECTION ACCEPTED

Mike Wege (Cure53): *I wonder if a runtime warning is appropriate. I am not sure how something like this could be accomplished - maybe with a debug message?*

CRL-01-010 Slashes and .. are decoded in file URIs (Low)

Daniel Stenberg (curl): Left without action. curl works on URLs and they can use percent-encoded characters.

REJECTION ACCEPTED

Mike Wege (Cure53): *I suggest that you further consider the security implications. Most likely a comment added to the documentation warning the user would suffice.*

CRL-01-012 Only the md5 of the SSH host key fingerprint is checked (Low)

Daniel Stenberg (curl): “Support better than MD5 hostkey hash” (for ssh) has been added as an item in the TODO list. That doesn’t mean that it will get implemented soon, but it at least means that it won’t be forgotten or has been ignored. The MD5 fingerprint is provided by libssh2, and it doesn’t support better hashes than SHA-1 which also is deprecated these days. We would ideally like a SHA-256 or something from libssh2.
https://curl.haxx.se/docs/todo.html#Support_better_than_MD5_hostkey

DEFERRAL ACCEPTED

CRL-01-015 Default Compile-time options lack support for PIE and RELRO (Low)

Added to TODO list:
https://curl.haxx.se/docs/todo.html#Enable_PIE_and_RELRO_by_default

DEFERRAL ACCEPTED

CRL-01-016 Unchecked snprintf() calls (Low)

Fix: https://github.com/curl/curl/commit/9885c9508ec757f7f658dab11658e4a3e643a420 (AddFormDataf() issue)

VERIFIED

Fix: https://github.com/curl/curl/commit/8238ba9c5f10414a88f502bf3f5d5a42d632984c (snprintf issue)

VERIFIED

CRL-01-017 Permit disabling (insecure) fallbacks (Low)

Fix: https://github.com/curl/curl/commit/ddefc056b68e0eb8912080b6817d17f1a8ad406f
Fix: https://github.com/curl/curl/commit/f682156a4fc6c43fb38db4abda49b9a1bc1ed368

Daniel Stenberg (curl): We will make sure that failing to seed the PRNG properly will cause an error at every call path that wants to use random. Only libcurl built completely without (a capable) TLS backend will resort to the non-crypto pseudo randomization.

VERIFIED

The mbedTLS backend still doesn't build with a good random source and we've posted a "cry for help" for it (which may or may not help).

DEFERRAL ACCEPTED

Mike Wege (Cure53): *This issue lacks an entry in the todo list. I think it would advantageous to create an entry for it, otherwise the issue may get lost over time.*

CRL-01-018 Null pointer dereference in the RTSP protocol (Low)

Fix: https://github.com/curl/curl/commit/8e8afa82cbb629bd2a95eba1cdf47f65dd62a6d5

VERIFIED

CRL-01-019 nss_init_sslver uses version info from NSS header (Info)

Fix: https://github.com/curl/curl/commit/curl-7_51_0-17-g5d45ced

VERIFIED

Mike Wege (Cure53): *We accept the deferral of setting the max supported version to TLS 1.3 until that version is stable.*

CRL-01-020 dup_nickname() doesn't check for memory allocation failure (Low)

Kamil Dudka (curl): There is no problem really. The code works as designed. Out of memory situations are handled sufficiently (although not explicitly). The code can be rewritten to be more readable but it is a low priority task.

NO FIX IMPLEMENTED

Mike Wege (Cure53): *I question the statement "works as designed". If it's been designed that way, then it has been designed rather poorly. Wrapping multiple meanings into one return value is considered error-prone coding style and should be avoided.*

Kamil Dudka (curl): I believe it is acceptable to return CURLE_SSL_CERTPROBLEM if libcurl is unable to load a client certificate. The fact that strdup() failed to allocate memory is an implementation detail. Returning CURLE_OUT_OF_MEMORY would be more precise but neither OOM failures inside NSS libraries, which are more likely to happen, are propagated like this. Compared to that, the failure of strdup() during nickname allocation is really a corner case.

CRL-01-022 polarssl_connect_step1() lacks matching unlock (Info)

Fix: https://github.com/curl/curl/commit/1e3161a20d5759409fec9aa339f79c5f71cabe65

VERIFIED

CRL-01-023 ssl_thread_setup() leaves mutex buffer partially uninitialised (Info)

Fix: https://github.com/curl/curl/commit/ace237ae4ee4a6399f2ab835aea937e6d4471d69

VERIFIED