NERDVANA / Crypt-SecretBuffer-0.019 / Changes

```
Version 0.019 - 2026-03-05
  - Method 'memcmp' and Span->cmp are now constant-time operations, protect
    against timing attacks when comparing passwords.
  - Method ->scan has a new CONST_TIME flag that improves protection agains
    timing attacks, but needs more work.
  - The ::PEM object has an improved design where ->header_kv is the offici
    array of attributes and 'headers' is a view of that array, providing
    features like multi-value, case-insensitivity, and whitespace trimming
    without modifying the original array.
  - Breaking Change: The return value of ->scan on failure to match is now
    empty list, where previously it was (x,0) where x was sometimes useful
    and sometimes not and would vary depending on other flags.

Version 0.018 - 2026-02-11
  - New exportable span() function which universally constructs a span
    from a SecretBuffer, Span object, or plain scalar.
  - PEM object header values are no longer secrets, by default.
    This is a breaking change, but the module has been out less than two mo
  - New methods to encode and decode variable-length integers and
    length-prefixed strings:
     - append_lenprefixed     / parse_lenprefixed
     - append_base128le       / parse_base128le
     - append_base128be       / parse_base128be
     - append_asn1_der_length / parse_asn1_der_length
  - New method Span->append_to, and normalize Span->copy_to to overwrite.
    This is a breaking change in that ->copy_to used to append to SecretBuf
    but overwrite scalars.  Now copy_to overwrites for both destination typ
    and append_to appends to both destination types.
  - Change type of secret_buffer_charset_test_codepoint codepoint from `uin
    to `U32` so that users of Inline aren't forced to `#include <stdint.h>`
    This is a breaking change for any C code which was looking up the funct
    name in the %C_API hash, but the ABI is unchanged.

Version 0.017 - 2025-12-28
  - New features for append_console_line: char_mask, char_count, char_class
  - New attribute 'line_input' for ConsoleState toggles line buffering
  - New method Span->set_up_us_the_bom processes byte-order-marks
  - Return value of memcmp is now normalized to -1/0/1

Version 0.016 - 2025-12-27
  - New append_console_line(prompt => $txt) feature resolves race condition
    between printing prompt and disabling echo
  - New Crypt::SecretBuffer::ConsoleState utility class lets users disable
    TTY echo more flexibly
  - Fix bug where ->sysread was actually calling the ->read implementation
  - Fix Win32 compatibility (compile errors in 0.013 - 0.015)
  - Fix TTY tests on BSD (race condition in append_console_line)

Version 0.015 - 2025-12-22
  - Fix 5.8 and C89 compatibility

```

```
    - New method $span->cmp($other) is like memcmp but on Unicode codepoints
    - Span objects now also have overloaded cmp, stringify, and boolean cast
    - New PEM parser class

Version 0.012 - 2025-12-19
    - String patterns passed to scanning functions (scan, index, rindex, and
      various methods of Span) can now be perl unicode strings, or SecretBuff
      objects, or Span objects (with their own encoding) and matches will be
      compared codepoint by codepoint.
    - Added BASE64 encoding
    - New exportable memcmp function, and method of SecretBuffer and Span obj
    - Exceptions in unmask_to and unmask_secrets_to now pass through to calle
      more correctly.
    - SecretBuffer overloads 'cmp' with the new memcmp function, so you can n
      compare buffers directly with cmp, lt, gt, eq and so on.
      (Span objects do not, yet, to avoid the complicated question of how bes
       compare mismatched encodings)
    - Fix debug-build-perl asertion failure in Span->copy_to when copying fro
      buffer of capacity=0

Version 0.011 - 2025-12-02
    - Fix bug in parsing strings of '$encoding' parameters that could return
      an uninitialized value for an invalid encoding.
    - Fix compatibility of tests for Win32 and Cygwin

Version 0.010 - 2025-11-22
    - Fix bug where substr with replacement that shrinks the buffer would lea
      un-wiped bytes at the end of the string.
    - Fix compile error on BSD, possibly others
    - New methods: splice, append
    - substr, splice, append, and assign now all have special cases for assig
      a Span of a buffer.
    - New C API functions:
       - secret_buffer_SvPVbyte
       - secret_buffer_splice
       - secret_buffer_splice_sv
    - Removed the short-lived secret_buffer_assign_sv from the previous relea
      since secret_buffer_splice_sv provides a more powerful way to do it.
    - C API is now exported as package variables for simpler conditional use
      the API, such as
      get_sv("Crypt::SecretBuffer::C_API::void secret_buffer_wipe(char *, siz
    - Standardized signatures of exported functions in these variables
    - Removed the macros
        SECRET_BUFFER_DECLARE_FUNCTION_POINTERS,
        SECRET_BUFFER_DEFINE_FUNCTION_POINTERS, and
        SECRET_BUFFER_DEFINE_FUNCTION_POINTERS
      because I'm pretty sure they'll never get used, and so now SecretBuffer
      doesn't need to include SecretBufferManualLinkage.h and the user can ju
      copy SecretBuffer.h as-is from the git repo.

Version 0.008 - 2025-11-17
    - New Span objects assist with parsing secrets.
    - New INI object for parsing common INI-like formats.
    - SecretBuffer now has 'rindex' and 'scan' methods, and 'index' allows th
      pattern to be a Regexp-ref as long as it is only a single character cla
    - New SecretBuffer methods load_file and save_file.
    - New C API functions:
       - secret_buffer_charset_from_regexpref
```

```
118             - secret_buffer_match_charset
119             - secret_buffer_match_bytestr
120             - secret_buffer_sizeof_transcode
121             - secret_buffer_transcode
122             - secret_buffer_assign_sv
123
124         Version 0.007 - 2025-10-31
125           - New API unmask_secrets_to and ->unmask_to
126           - Documented ways to use SecretBuffer without depending on it
127
128         Version 0.006 - 2025-09-09
129           - Fix bug checking errno for EAGAIN after failed _append_random
130           - Fix Makefile.PL feature tests for required lib linker flags
131           - Fix Win32 compat when wincrypt.h isn't required/available
132
133         Version 0.005 - 2025-06-05
134           - Fix bug in index() that fails to search on final char of buffer
135           - Improved OS feature detection, which should fix many build failures
136           - Fix 5.8 compat
137           - More unit tests
138           - Document security policy
139
140         Version 0.004 - 2025-06-05
141           - Add "Inline with => Crypt::SecretBuffer" support
142           - Tooling now provides C API from the XS shared lib as native exported sy
143           - Fix failing assertion in substr()
144
145         Version 0.003 - 2025-05-23
146           - Make attribute accessor for stringify_mask
147             Also allows stringify_mask to be specified in the constructor
148
149         Version 0.002 - 2025-05-23
150           - Fix 5.8 compatibility
151           - Fix exports & documentation
152
153         Version 0.001 - 2025-05-23
154           - Initial version, working on Linux, FreeBSD, and MSWin32
```