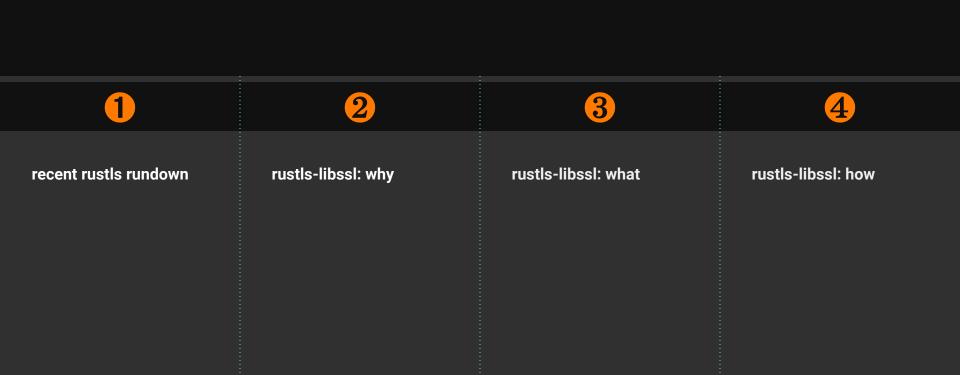


# Replacing OpenSSL, one step at a time

RustNL 2024

# this talk



#### recent rustls rundown

Funding!

Joe <u>@ctz</u> and Daniel <u>@cpu</u> full time

Adolfo <u>@aochagavia</u> and Ferrous Systems project contracts

Funded by ISRG Prossimo project: <a href="mailto:memorysafety.org">memorysafety.org</a>

#### recent rustls rundown

Shipped features!

Revocation support with CRLs

Pluggable cryptography providers

FIPS140 support

no\_std support

Post-quantum cryptography support

Unbuffered API

experimental



# rustls-libssl: why

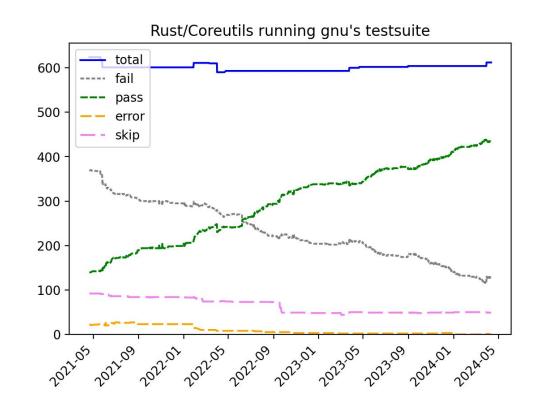
non-Rust users also deserve memory-safe TLS!

# rustls-libssl: why

replacing a memory-unsafe component with a drop-in alternative is a quick way to reduce risks

rustls-lib

replacedrop-

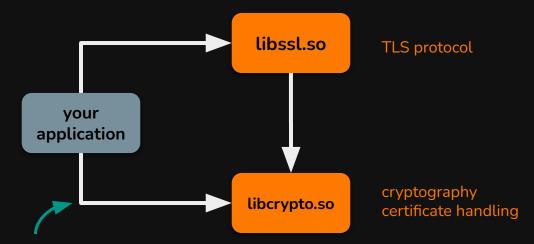


sks

source: <a href="https://github.com/uutils/coreutils">https://github.com/uutils/coreutils</a>

#### rustls-libssl: what

### OpenSSL is **two** libraries:



Invariably an application that uses libssl also uses libcrypto

#### rustls-libssl: what

OpenSSL is two libraries:

dynamically linked

TLS protocol

cryptography
certificate handling

#### rustls-libssl: what

if we implement the same ABI, we can replace just this OpenSSL is **two** libraries: libssl.so TLS protocol your application cryptography libcrypto.so certificate handling

# OpenSSL libssl is

#### 522 functions!

BIO\_f\_ssl BIO new buffer ssl connect BIO\_new\_ssl BIO\_ssl\_copy\_session\_id BIO\_ssl\_shutdown d2i SSL SESSION DTLS client method DTLS\_get\_data\_mtu DTLSv1\_2\_client\_method DTLSv1\_2\_method DTLSv1 2 server method DTLSv1\_client\_method DTLSv1\_listen DTLSv1\_method i2d SSL SESSION OPENSSL init ssl PEM read bio SSL SESSION PEM\_write\_bio\_SSL\_SESSION PEM\_write\_SSL\_SESSION SSL\_accept SSL add1 host SSL\_add1\_to\_CA\_list SSL add file cert subjects to stack SSL add store cert subjects to stack SSL\_alert\_type\_string SSL\_alert\_type\_string\_long SSL\_alloc\_buffers SSL bytes to cipher list SSL callback ctrl SSL check chain SSL\_check\_private\_key SSL CIPHER get auth nid SSL CIPHER get cipher nid SSL\_CIPHER\_get\_handshake\_digest SSL\_CIPHER\_get\_id SSL\_CIPHER\_get\_kx\_nid SSL CIPHER get version SSL\_clear\_options SSL\_client\_hello\_get0\_ciphers SSL client hello get0 compression methods

SSL\_client\_hello\_get0\_legacy\_version SSL\_client\_hello\_get0\_random

SSL\_COMP\_add\_compression\_method SSL\_COMP\_get0\_name SSL\_COMP\_get\_compression\_methods SSL\_COMP\_get\_id SSL COMP set0 compression methods SSL\_CONF\_cmd\_value\_type SSL\_CONF\_CTX\_clear\_flags SSL\_CONF\_CTX\_finish SSL\_CONF\_CTX\_free SSL CONF CTX new SSL CONF CTX set flags SSL\_config SSL connect SSL\_copy\_session\_id SSL ct is enabled SSL CTX add1 to CA list SSL\_CTX\_add\_custom\_ext SSL CTX add session SSL CTX check private key SSL\_CTX\_clear\_options SSL\_CTX\_config SSL CTX ct is enabled SSL CTX dane clear flags SSL CTX dane enable SSL\_CTX\_flush\_sessions SSL\_CTX\_free SSL\_CTX\_get0 CA list SSL CTX get0 certificate SSL CTX get0 param SL\_CTX\_get0\_privatekey SSL\_CTX\_get0\_security\_ex\_data SSL CTX get cert store SSL CTX get client cert cb SSL CTX get num tickets SSL CTX get quiet shutdown SSL\_CTX\_get\_record\_padding\_callback\_arg SSL CTX get security callback SSL CTX get security level SSL\_CTX\_get\_ssl\_method SSL CTX get timeout SSL CTX get verify mode

SSL\_CTX\_has\_client\_custom\_ext SSL\_CTX\_load\_verify\_dir SSL CTX new SSL CTX new ex SSL CTX remove session SSL\_CTX\_sess\_get\_remove\_cb SSL\_CTX\_sessions SSL\_CTX\_sess\_set\_get\_cb SSL CTX sess set remove cb SSL\_CTX\_set0\_CA\_list SSL\_CTX\_set0\_ctlog\_store SSL\_CTX\_set0\_security\_ex\_data SSL CTX set1 param SSL CTX set allow early data cb SSL CTX set alon protos SSL\_CTX\_set\_async\_callback\_arg SSL\_CTX\_set\_block\_padding SSL CTX set cert verify callback SSL\_CTX\_set\_client\_cert\_cb SSL\_CTX\_set\_client\_cert\_engine SSL\_CTX\_set\_default\_ctlog\_list\_file SSL\_CTX\_set\_default\_passwd\_cb SSL\_CTX\_set\_default\_passwd\_cb\_userdata SSL\_CTX\_set\_default\_read\_buffer\_len SSL CTX set generate session id SSL CTX set info callback SSL CTX set max early data SSL\_CTX\_set\_next\_protos\_advertised\_cb SSL\_CTX\_set\_next\_proto\_select\_cb SSL\_CTX\_set\_not\_resumable\_session\_callback SSL\_CTX\_set\_num\_tickets SSL CTX set post handshake auth SSL\_CTX\_set\_psk\_client\_callback SSL\_CTX\_set\_psk\_find\_session\_callback SSL\_CTX\_set\_psk\_server\_callback SSL CTX set quiet shutdown SSL\_CTX\_set\_record\_padding\_callback SSL CTX set record padding callback arg SSL\_CTX\_set\_recv\_max\_early\_data

SSL CTX set security level

SSL\_CTX\_set\_srp\_cb\_arg SSL\_CTX\_set\_srp\_client\_pwd\_callback SSL\_CTX\_set\_srp\_password SSL CTX set srp username callback SSL\_CTX\_set\_srp\_verify\_param\_callback SSL\_CTX\_set\_timeout SSL\_CTX\_set\_tlsext\_max\_fragment\_length SSL CTX set tlsext ticket key evp cb SSL CTX set tmp dh callback SSL\_CTX\_set\_verify SSL\_CTX set verify depth SSL CTX SRP CTX init SSL CTX up ref SSL\_CTX\_use\_certificate\_ASN1 SSL\_CTX\_use\_certificate\_chain\_file SSL\_CTX\_use\_certificate\_file SSL\_CTX\_use\_PrivateKey SSL CTX use PrivateKey ASN1 SSL CTX use psk identity hint SSL\_CTX\_use\_RSAPrivateKey SSL\_CTX\_use\_RSAPrivateKey\_ASN1 SSL\_CTX\_use\_RSAPrivateKey\_file SSL CTX use serverinfo SSL CTX use serverinfo file SSL\_dane\_set\_flags SSL\_dane\_tlsa\_add SSL\_do\_handshake SSL export keying material SSL\_export\_keying\_material\_early SSL\_free\_buffers SSL\_get0\_alpn\_selected SSL\_get0\_CA\_list SSL\_get0\_dane SSL\_get0\_dane\_authority SSL\_get0\_dane\_tlsa SSL\_get0\_next\_proto\_negotiated SSL\_get0\_param SSL\_get0\_peer\_CA\_list SSL get0 peer certificate SSL\_get0\_peername SSL\_get0\_peer\_scts SSL\_get0\_security\_ex\_data SSL get1 session

SSL\_get\_cipher\_list SSL\_get\_ciphers SSL\_get\_client\_CA\_list SSL get current cipher SSL\_get\_current\_compression SSL get current expansion SSL\_get\_default\_passwd\_cb\_userdata SSL\_get\_default\_timeout SSL\_get\_early\_data\_status SSL get ex data SSL\_get\_ex\_data\_X509\_STORE\_CTX\_idx SSL\_get\_fd SSL\_get\_finished SSL\_get\_info\_callback SSL\_get\_max\_early\_data SSL get num tickets SSL\_get\_peer\_signature\_type\_nid SSL\_get\_pending\_cipher SSL\_get\_privatekey SSL get psk identity SSL get psk identity hint SSL\_get\_quiet\_shutdown SSL\_get\_rbio SSL\_get\_read\_ahead SSL\_get\_security\_callback SSL\_get\_servername\_type SSL\_get\_server\_random SSL\_get\_session SSL get shared ciphers SSL get shared sigalgs SSL\_get\_shutdown SSL\_get\_sigalgs SSL\_get\_signature\_type\_nid SSL get srp userinfo SSL\_get\_srtp\_profiles SSL\_get\_SSL\_CTX SSL\_get\_ssl\_method SSL\_get\_state SSL\_get\_verify\_callback SSL\_get\_version SSL has matching session id SSL\_has\_pending SSL\_in\_before SSL in init SSL is server

SSL\_key\_update SSL load client CA file SSL load client CA file ex SSL\_peek\_ex SSL\_pending SSL\_read SSL\_read\_early\_data SSL\_read\_ex SSL\_renegotiate SSL\_renegotiate\_abbreviated SSL\_renegotiate\_pending SSL rstate string SSL select next proto SSL\_sendfile SSL SESSION get0 alpn selected SSL SESSION get0 hostname SSL\_SESSION\_get0\_id\_context SSL\_SESSION\_get0\_ticket SSL\_SESSION\_get0\_ticket\_appdata SSL\_SESSION\_get\_compress\_id SSL\_SESSION\_get\_ex\_data SSL\_SESSION\_get\_master\_key SSL\_SESSION\_get\_max\_early\_data SSL\_SESSION\_get\_max\_fragment\_length SSL\_SESSION\_get\_protocol\_version SSL\_SESSION\_get\_ticket\_lifetime\_hint SSL\_SESSION\_get\_time SSL SESSION has ticket SSL\_SESSION\_print\_keylog SSL session reused SSL SESSION set1 alpn selected SSL\_SESSION\_set1\_hostname SSL\_SESSION\_set1\_id SSL\_SESSION\_set1\_id\_context SSL SESSION set cipher SSL SESSION set max early data SSL SESSION set timeout SSL\_set0\_CA\_list SSL\_set0\_tmp\_dh\_pkey SSL\_set0\_wbio SSL\_set1\_param SSL set accept state SSL\_set\_alpn\_protos SSL set async callback

SSL\_set\_cipher\_list SSL set ciphersuites SSL\_set\_client\_CA\_list SSL\_set\_connect\_state
SSL\_set\_ct\_validation\_callback SSL set debug SSL\_set\_default\_passwd\_cb SSL\_set\_ex\_data SSL\_set\_generate\_session\_id SSL\_set\_hostflags SSL set info callback SSL set max early data SSL set not resumable session callback SSL\_set\_num\_tickets SSL\_set\_options SSL set post handshake auth SSL set psk find session callback SSL set psk server callback SSL\_set\_quiet\_shutdown SSL\_set\_read\_ahead SSL\_set\_record\_padding\_callback SSL set record padding\_callback arg SSL set recv max early data SSL\_set\_rfd SSL\_set\_security\_callback SSL\_set\_security\_level SSL\_set\_session SSL set session id context SSL set session ticket ext SSL set session ticket ext cb SSL\_set\_srp\_server\_param\_pw SSL\_set\_SSL\_CTX SSL\_set\_ssl\_method SSL set tlsext max fragment length SSL set tlsext use srtp SSL\_set\_tmp\_dh\_callback SSL\_set\_trust SSL\_set\_verify\_depth SSL set verify result SSL\_shutdown SSL\_SRP\_CTX\_free SSL\_srp\_server\_param\_with\_username SSL\_state\_string\_long SSL test functions SSL use cert and key SSL\_use\_certificate SSL\_use\_certificate\_ASN1 SSL use certificate chain file SSL use PrivateKev ASN1 SSL\_use\_RSAPrivateKey SSL\_use\_RSAPrivateKey\_ASNI

SSL use REAPrivateley\_file
SSLV2\_client\_method
SSLV3\_client\_method
SSLV3\_server\_method
SSLV\_serier\_client\_post\_handshake
SSL\_verier\_client\_post\_handshake
SSL\_verier\_client\_post\_handshake
SSL\_verier\_client\_post\_handshake
SSL\_verier\_client\_post\_handshake
SSL\_verier\_client\_post\_handshake
SSL\_verier\_client\_post\_handshake
SSL\_verier\_client\_post\_handshake
SSL\_verier\_client\_post\_handshake
SSL\_verier\_method
TSLS\_verier\_method
TSLS\_verier\_method
TSLS\_verier\_method
TSLS\_verier\_client\_post\_handshake
TSLS\_verier\_method
TSLS\_verier\_method
TSLS\_verier\_method
TSLS\_verier\_method
TSLS\_verier\_method
TSLS\_verier\_method
TSLS\_verier\_method

#### 136 additional functions are multiplexed!

(SSL\_CTX\_ctrl, SSL\_ctrl, SSL\_CTX\_callback\_ctrl and SSL\_callback\_ctrl are magic like this.)

#### Fortunately, most of this API surface is rarely used.

# eg. curl<sup>1</sup> uses just 57 functions:

```
BIO f ssl
                                         SSL CTX set keylog callback
                                                                                   SSL get peer cert chain
                                         SSL CTX set msg callback
                                                                                   SSL get privatekey
OPENSSL init ssl
SSL_alert_desc_string_long
                                         SSL_CTX_set_next_proto_select_cb
                                                                                   SSL get shutdown
SSL CIPHER get name
                                         SSL CTX set options
                                                                                   SSL get verify result
SSL connect
                                         SSL CTX set post handshake auth
                                                                                   SSL get version
                                         SSL_CTX_set_srp_password
SSL ctrl
                                                                                   SSL new
SSL CTX add client CA
                                         SSL CTX set srp username
                                                                                   SSL pending
SSL_CTX_check_private_key
                                         SSL CTX set verify
                                                                                   SSL read
                                         SSL CTX use certificate chain file
                                                                                   SSL SESSION free
SSL CTX ctrl
                                         SSL CTX use certificate file
SSL CTX free
                                                                                   SSL set bio
                                         SSL CTX use certificate
SSL CTX get cert store
                                                                                   SSL set connect state
SSL CTX load verify dir
                                         SSL CTX use PrivateKey file
                                                                                   SSL set ex data
SSL_CTX_load_verify_file
                                         SSL_CTX_use_PrivateKey
                                                                                   SSL_set_fd
                                         SSL free
SSL_CTX_new
                                                                                   SSL_set_session
SSL CTX sess set new cb
                                         SSL get0 alpn selected
                                                                                   SSL shutdown
                                         SSL get1 peer certificate
SSL CTX set alpn protos
                                                                                   SSL write
                                         SSL get certificate
                                                                                   TLS client method
SSL CTX set cipher list
SSL_CTX_set_ciphersuites
                                         SSL get current cipher
SSL_CTX_set_default_passwd_cb
                                         SSL_get_error
                                         SSL get_ex_data
SSL CTX set default passwd cb userdata
```

<sup>&</sup>lt;sup>1</sup> curl 7.81.0-1ubuntu1.15 as shipped on Ubuntu 22.04LTS

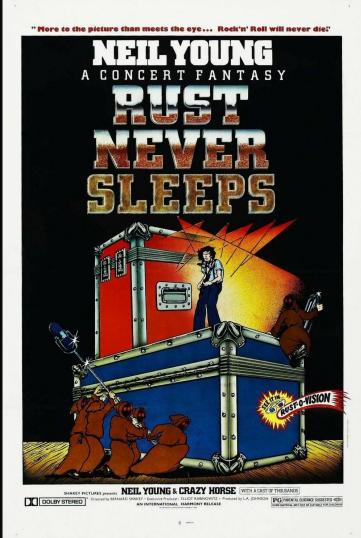
nginx<sup>1</sup> uses 90 functions.

significant overlap with curl: 114 functions cover both

that feels achievable!

<sup>&</sup>lt;sup>1</sup> nginx 1.18.0-6ubuntu14.4 as shipped on Ubuntu 22.04LTS

let's make libssl.so (but in rust)





build a C-ABI dynamic library

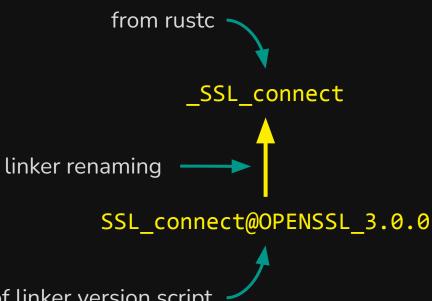
#### Cargo.toml:

```
[lib]
name = "ssl"
crate-type = ["cdylib"]
```

- build a C-ABI dynamic library
- with correctly-versioned symbols

```
build.rs:
let filename = write_version_file();
println!("cargo:rustc-cdylib-link-arg="
         "-Wl,--version-script={filename}");
for symbol in ENTRYPOINTS {
    println!(
        "cargo:rustc-cdylib-link-arg="
        "-Wl,--defsym={}=_{}",
        symbol, symbol
(and then use lld)
```

- build a C-ABI dynamic library
- with correctly-versioned symbols



subject of linker version script

- build a C-ABI dynamic library
- with correctly-versioned symbols
- write some rust functions with C linkage

```
build a C-ABI dvnamic librarv

#[no_mangle] and extern "C"

entry! {
    pub fn _SSL_alert_desc_string(value: c_int) -> *const c_char {
        crate::constants::alert_desc_to_short_string(value).as_ptr() as *const c_char
    }
}
```

- build a C-ABI dynamic library
- with correctly-versioned symbols
- write some rust functions with C linkage
- and avoid undefined behaviour!

```
build a C-ABI dvnamic librarv

also wrap the whole function body in
std::panic::catch_unwind

entry! {
    pub fn _SSL_alert_desc_string(value: c_int) -> *const c_char {
        crate::constants::alert_desc_to_short_string(value).as_ptr() as *const c_char
    }
}
```

write some simple C programs that use OpenSSL

```
#include <stdio.h>
#include <openssl/ssl.h>
int main(void) {
  for (int i = -1; i < 260; i++) {
    printf("%d: '%s' '%s'\n", i,
           SSL alert desc string(i),
           SSL_alert_desc_string_long(i));
  return 0;
```

- write some simple C programs that use OpenSSL
- have them print all return values and data

```
-1: 'UK' 'unknown'
0: 'CN' 'close notify'
1: 'UK' 'unknown'
2: 'UK' 'unknown'
3: 'UK' 'unknown'
4: 'UK' 'unknown'
5: 'UK' 'unknown'
6: 'UK' 'unknown'
7: 'UK' 'unknown'
```

should be deterministic: so no pointer values, or randomised data!

- write some simple C programs that use OpenSSL
- have them print all return values and data
- run them against the real libssl.so and ours

```
$ ./target/constants > original
$ LD_LIBRARY_PATH=target/debug
    ./target/constants > ours
```

- write some simple C programs that use OpenSSL
- have them print all return values and data
- run them against the real libssl.so and ours
- the output should be identical!

```
$ ./target/constants > original

$ LD_LIBRARY_PATH=target/debug
   ./target/constants > ours

$ diff -su original ours
Files original and ours are identical
```

miri works very well for pure-rust libraries

...but not if you make an FFI call

miri works very well for pure-rust libraries

...but not if you make an FFI call

miri works very well for pure-rust libraries

...but not if you make an FFI call

```
error: unsupported operation: can't call foreign

--> src/x509.rs:242:27

242 | raw: unsafe { X509_STORE_new() },

^^^^^^^^^^^^^^^^ can't conforming function

X509_STORE_new` on OS `linux`

= help: this is likely not a bug in the program; it indicates that the program
```

Let's send a

pull request

= help: this is likely not a bug in the program; it indicates that the program
performed an operation that the interpreter does not support

miri already looks in your crate for native function definitions!

```
#[cfg(miri)]
mod miri {
    pub struct X509_STORE(());
    #[no_mangle]
    pub extern "C" fn X509_STORE_new() -> *mut X509_STORE {
        Box::into_raw(Box::new(X509_STORE(())))
    #[no_mangle]
    pub extern "C" fn X509_STORE_free(ptr: *mut X509_STORE) {
        if ptr.is_null() { return; }
        drop(unsafe { Box::from_raw(ptr) });
```



we've made a memory-safe<sup>1</sup> replacement for openssl 3.0 libssl.so



we've made a memory-safe<sup>1</sup> replacement for openssl 3.0 libssl.so



written in rust, using rustls

- ended with the wealth of the w
- mritten in rust, using rustls
- applications don't need recompilation

- we've made a memory-safe<sup>1</sup> replacement for openssl 3.0 libssl.so
- written in rust, using rustls
- applications don't need recompilation
- supports a small subset of the openssl 3.0 libssl API

(but enough for nginx and curl<sup>2</sup>)

mostly
 ubuntu 22.04 LTS versions

```
and it's easy to try:
```

```
$ with-rustls-libssl curl https://rustls.horse/
or

# rustls-libssl-nginx enable
# systemctl daemon-reload
# service nginx restart
```

#### thanks!

Repo: <a href="https://github.com/rustls/rustls-openssl-compatigital">https://github.com/rustls/rustls-openssl-compatigital</a>

Mastodon: @jpixton@octodon.social

Mail: jbp@jbp.io

Slides: <a href="https://github.com/ctz/talks">https://github.com/ctz/talks</a>