Forensic Tool Development with Rust

Case study: stringsext

Author: Dipl.-Ing. Jens Getreu
Supervisor: Prof. Olaf Manuel Maennel

Forensic-Tool Requirements

- Hard disk/memory images are huge

  → code efficiency!

- Images may contain malicious code exploiting potential code vulnerabilities

  → memory safety

Rust Programming Language

code efficiency

- zero cost abstractions
- no garbage collector

memory safety

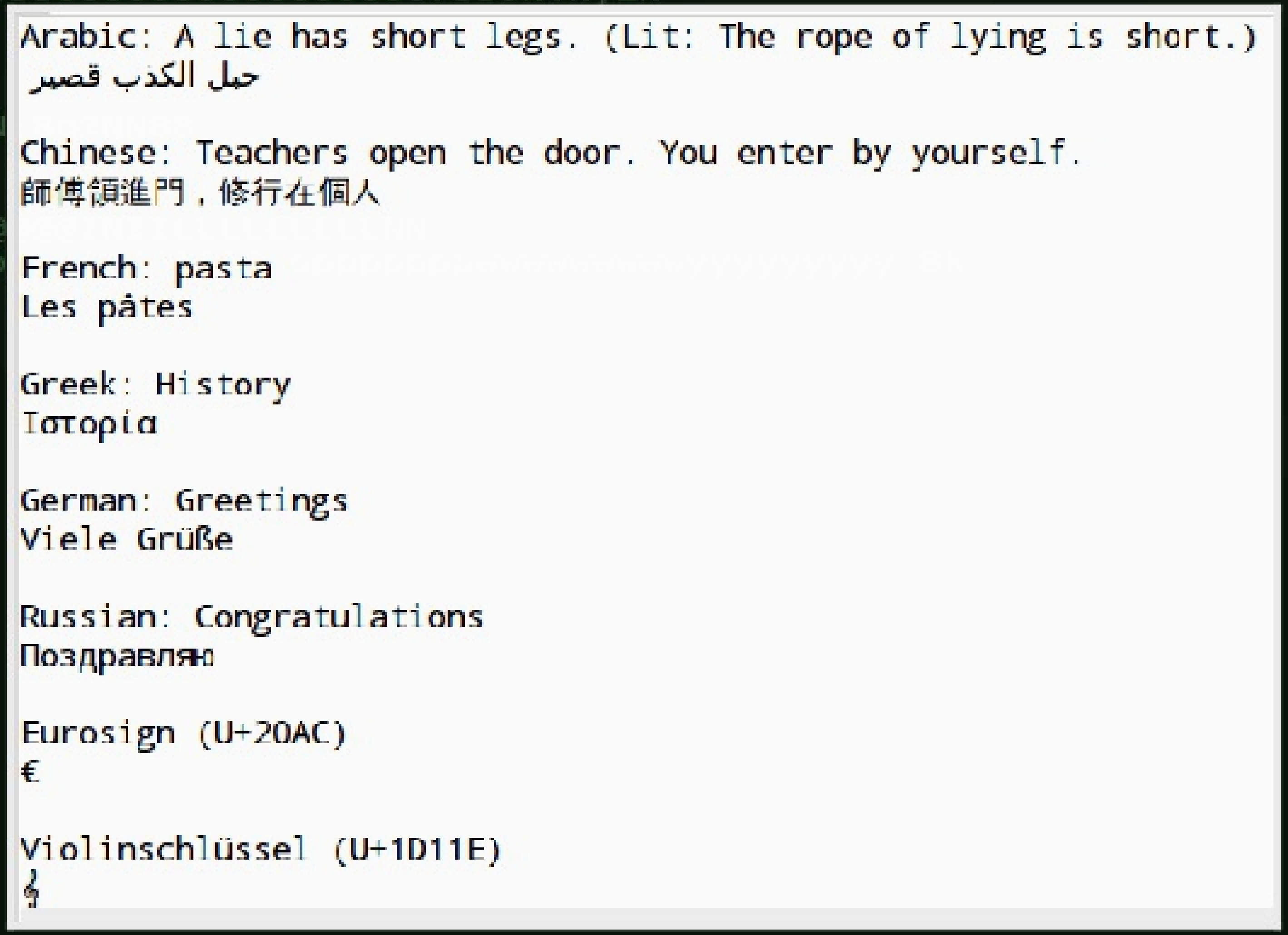
- data ownership

Hypothesis

Is the Rust ecosystem suitable for forensic tool development?

Method: Case study: Development of a GNU-strings alternative with multi-byte-encoding support

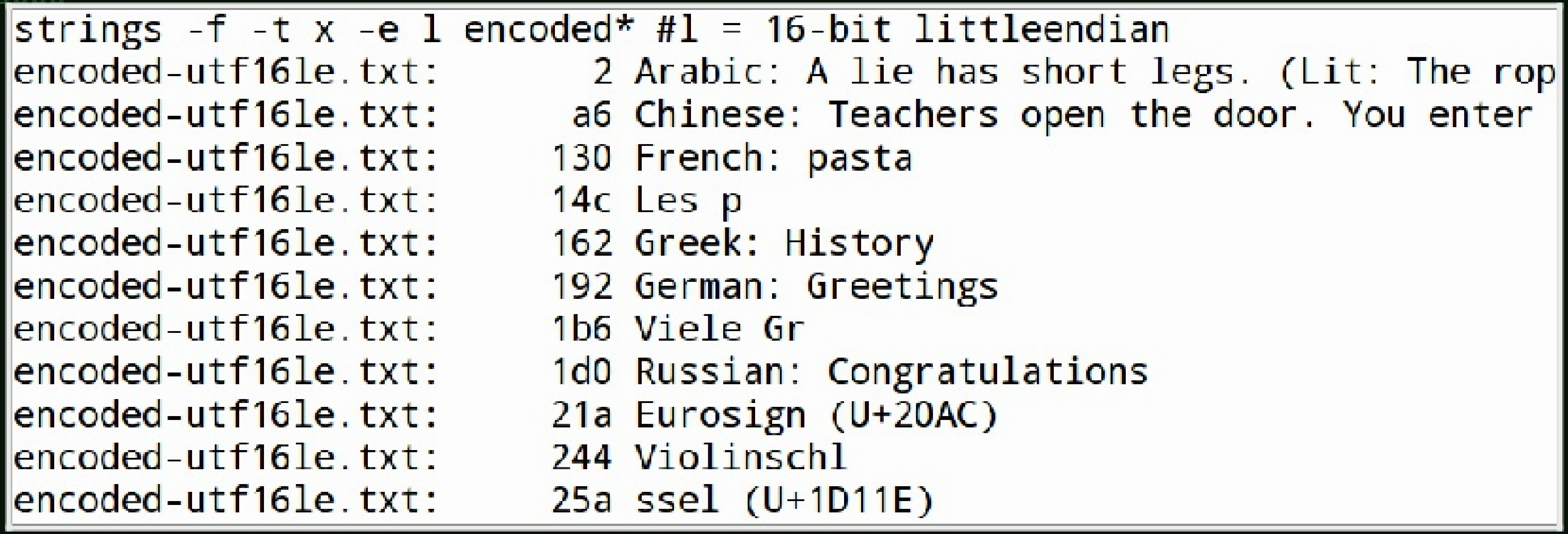
Test Case



```
Arabic: A lie has short legs. (Lit: The rope of lying is short.)
حبل الكذب قصير

Chinese: Teachers open the door. You enter by yourself.
師傅領進門，修行在個人

French: pasta
Les pâtes

Greek: History
Ιστορία

German: Greetings
Viele Grüße

Russian: Congratulations
Поздравляю

Eurosign (U+20AC)
€

Violinschlüssel (U+1D11E)
𝄞
```

(UTF-16LE encoded)

GNU-strings: Output



```
strings -f -t x -e l encoded* #l = 16-bit littleendian
encoded-utf16le.txt:       2 Arabic: A lie has short legs. (Lit: The rop
encoded-utf16le.txt:      a6 Chinese: Teachers open the door. You enter
encoded-utf16le.txt:     130 French: pasta
encoded-utf16le.txt:     14c Les p
encoded-utf16le.txt:     162 Greek: History
encoded-utf16le.txt:     192 German: Greetings
encoded-utf16le.txt:     1b6 Viele Gr
encoded-utf16le.txt:     1d0 Russian: Congratulations
encoded-utf16le.txt:     21a Eurosign (U+20AC)
encoded-utf16le.txt:     244 Violinschl
encoded-utf16le.txt:     25a ssel (U+1D11E)
```

Very limited multi-byte-encoding support

```
ABCNDAEN<AAA
AAstAAAssuAAAuuuuuuuuuuuuuuuuuvAA8N¥98:N;8q@NN88
:N;8¤©NN88N8<8
AAAAA£AAA££££££££££££££££AA¢N
,,,,,,,,,1111111118kN8888>>>>>>>>>>@@@@@@@@@INIILLLLLLLLLLNN
PPbbbNb8bbbcccccccccceezBeeeeejjjjj=jjjoQoAAAAAAAAAppppppppppwwwwwwwwwyyyyyyyyy|8k||||
¬¬¬¬¬¬¬¬¬¬¬¬¬¬¬¬¬¬¬¬¬¬¬¬¬¬¬

ĀĀĀĀĀĀĀĀĀĀĀĀĀĀĀĀĀĀĀĀĀĀĀĀĀĀĀ
```

"Don't run strings on untrusted files."

The setup_group function in bfd/elf.c in libbfd in GNU binutils 2.24 and earlier allows remote attackers to cause a denial of service (crash) and possibly execute arbitrary code via crafted section group headers in an ELF file.

—CVE-2014-8485

```
Š

ЂAЂBЂCЂDЂWЂXЂ
ЂMЂKЂPЂHЂ


ä

yYuUiIoOpP

·ʷ3¸

·ʷĬĸ

LMNOPQRSTUVWXYZ[\]^_`abcdefghijklmnopqrstuvwxy

LMNOPQRSTUVWXYZ[\]^_`abcdefghijklmnopqrstuvwxy

llocate variable buffer
o relocation entry
 overflows binary
 overflows binary
llocate memory for hash context
nitialise hash
enerate hash
inary: %x %x
llocate section header
ection header
```

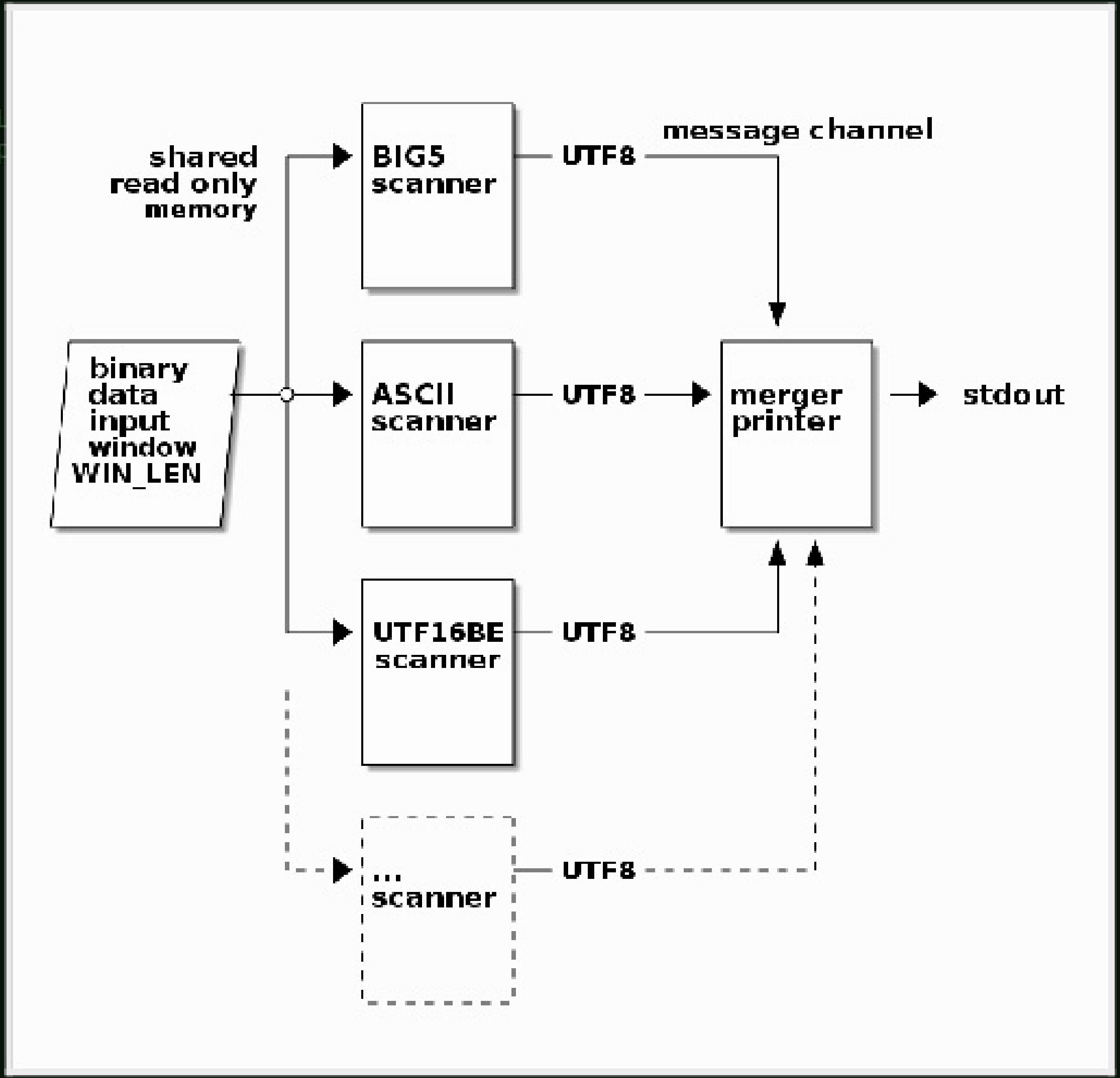
Stringsext

- a GNU-strings alternative with multi-byte-encoding support for
  - UTF-8, UTF-16be, UTF-16le, BIG5-2003, EUC-JP, KOI8-R and many others

Contribution

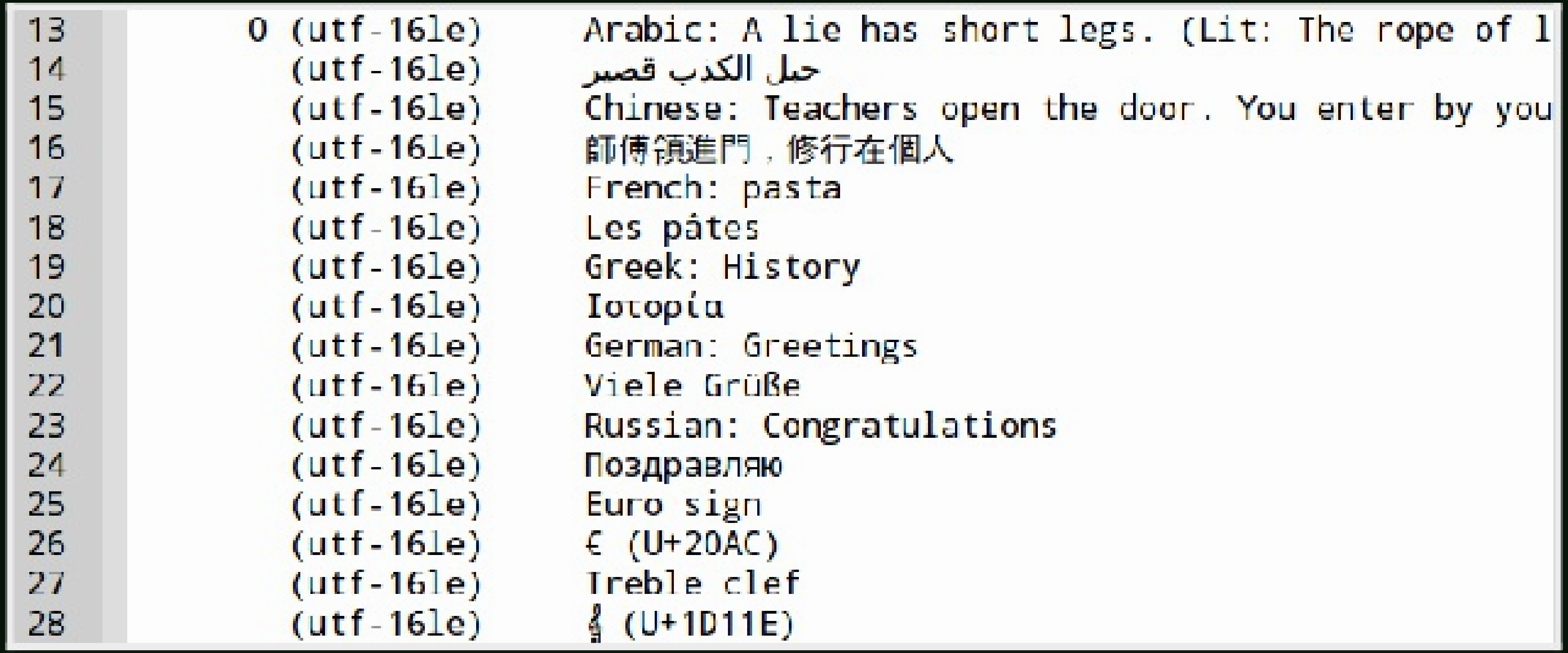
- production software stringsext (free source code)
- automated tests: 90 assertions
- 6 automated benchmarks and 2 automated field experiments
- user documentation
- 87 html pages of `developer documentation`_
- production builds for: Linux 32 bit, Linux 64 bit, Windows 32 bit, Windows 64 bit

```
¬¬¬¬¬¬¬¬¬¬¬¬¬¬¬¬¬¬
-+lNNU\0?`V033334333453334463336666666666666667339N:N;ZN
:F;W]^NXN88Y<RfgNh
N
A
AAuuuuuuuuuuuuuuuvAA8N¥98:N;8q@NN88
8<8
££££££££££££££££AA¢N
11111118kN8888>>>>>>>>>@@@@@@@@@INIILLLLL
cccccccccceezBeeeeejjjjj=jjjoQoAAAAAAAAAp
¬¬¬¬¬¬¬¬¬¬¬¬¬¬¬¬¬

ĀĀĀĀĀĀĀĀĀĀĀĀĀĀĀĀĀ

Xƀ

WXYZ[\]^_`abcdefghijklmnopqrstuvwxy

WXYZ[\]^_`abcdefghijklmnopqrstuvwxy

iable buffer
n entry
binary
binary
ory for hash context
ash
```



Design

Stringsext Output



```
      0 (utf-16le)     Arabic: A lie has short legs. (Lit: The rope of l
        (utf-16le)     حبل الكذب قصير
        (utf-16le)     Chinese: Teachers open the door. You enter by you
        (utf-16le)     師傅領進門，修行在個人
        (utf-16le)     French: pasta
        (utf-16le)     Les pâtes
        (utf-16le)     Greek: History
        (utf-16le)     Ιστορία
        (utf-16le)     German: Greetings
        (utf-16le)     Viele Grüße
        (utf-16le)     Russian: Congratulations
        (utf-16le)     Поздравляю
        (utf-16le)     Euro sign
        (utf-16le)     € (U+20AC)
        (utf-16le)     Treble clef
        (utf-16le)     𝄞 (U+1D11E)
```

Demonstration

Combine UTF-16 Little-Endian and Big-Endian scanning and prevent false positives:

```
cat /dev/sda2  | ./stringsext -ci -tx \
            -e UTF-16be,16,U+0..U+007f \
            -e UTF-16be,30,U+20..U+2f,U+400..U+07ff \
            -e UTF-16le,10,U+0..U+007f \
            -e UTF-16le,30,U+20..U+2f,U+400..U+07ff \
```

Results

- [-] more bugs in young Rust libraries, but
- [+] Rust is memory safe and
- [+] bugs are much easier to detect than memory safety related vulnerabilities

→ Rust meets the requirements of forensic tool development