HANDS-ON NETWORK PROGRAMMING WITH C

Written By Lewis Van Winkle

ERRATA

Unfortunately, it was inevitable that some mistakes made it into Hands-On Network Programming with C.

This page lists the corrections I've identified so far. If you've noticed a mistake that isn't listed here, please contact me so I can update this page.

(All pages numbers refer to the print edition.)

ERRATA FOR HANDS-ON NETWORK PROGRAMMING WITH C

PAGE 38

In the `unix_list.c` code example, `address->ifa_addr` should be checked to ensure that it is not a null pointer. If null, set `address = address->ifa_next` and `continue`.

In other words, the `while` loop needs the following check:

```
if (address->ifa_addr == NULL) {
    address = address->ifa_next;
    continue;
}
```

You can view the corrected example in the GitHub repository here.

PAGE 46

There is a typo on page 46. The passage "for severing web pages" should be "for *serving* web pages."

PAGE 79

There is a typo on page 79. The code function `FD_ISSSET` should be `FD_ISSET`.

PAGE 134

There is a typo on page 134. `getaddrinof()` should be `getaddrinfo()`.

PAGE 152

The code to read `rcode` uses the wrong bitmask. The code

```
const int rcode = msg[3] & 0x07;
```

should be replaced with

```
const int rcode = msg[3] & 0x0F;
```

You can view the corrected example in the GitHub repository here.

PAGE 155

The size of the TTL field is stated incorrectly. The text reading "a 16-bit TTL field" should read "a 32-bit TTL field." The example code works on a 32-bit field and is correct.

PAGE 157

The code that checks for domain names exceeding 255 characters, should be checking for domain names exceeding 253 characters. The DNS standard limits the total number of bytes representing a domain name (including label lengths and the optional dot at the end) to 255, which leaves only 253 characters for the domain name itself. See here for more info.

PAGE 182

The code at the top of the page should start with `++p;` directly before the `while` statement. Without this, the `#`, if it exists, will not be found and removed from the URL.

PAGE 220

There is a mistake in the `web_server.c` example code on page 220.

The code:

```
if (MAX_REQUEST_SIZE == client->received) {
    send_400(client);
```

```
    continue;
}
```

Should be:

```
if (MAX_REQUEST_SIZE == client->received) {
    send_400(client);
    client = next;
    continue;
}
```

You can view the corrected example in the GitHub repository here.

PAGE 267

The sentence reading "However, deriving one key from the other after the fact is not possible." is incorrect. For many asymmetric ciphers, it is easy to derive the public key from the private key. Deriving the private key from the public key is computational intractable.

PAGE 275

There is a mistake in the code to initialize a TLS connection.

The code:

```
SSL *ssl = SSL_new(ctx);
if (!ctx) {
    ...
```

Should be:

```
SSL *ssl = SSL_new(ctx);
if (!ssl) {
    ...
```

This error is repeated on pages 280, 295, and 299. The code examples in the GitHub repository have been corrected.

PAGE 367

The sentence that reads, "In step 5, setting the socket back to non-blocking mode," should actually say "back to *blocking* mode".

PAGE 383

It should be noted the using the `SO_REUSEADDR` flag on Windows will allow multiple programs to bind to the same port. This can certainly cause issues!

PAGE 434

You may run into trouble when compiling libssh on Windows with MinGW. When compiling libssh you may see an error stating "Your system must have getaddrinfo()." This can be resolved by taking the following steps:

Open the `src/connect.c` file in a text editor. Go to the line that's causing the error. It's line 88 in the libssh 0.9.2 version.

You'll see this code:

```
#ifndef HAVE_GETADDRINFO
#error "Your system must have getaddrinfo()"
#endif
```

Just comment-out or delete those three lines.

There is another issue in `src/CMakeLists.txt` which will trip you up. Open that file. You'll see this on line 10:

```
if (WIN32)
  set(LIBSSH_LINK_LIBRARIES
    ${LIBSSH_LINK_LIBRARIES}
    ws2_32
  )
endif (WIN32)
```

You need to add `crypto` after `ws2_32`. In other words, it should look like this:

```
if (WIN32)
  set(LIBSSH_LINK_LIBRARIES
    ${LIBSSH_LINK_LIBRARIES}
    ws2_32 crypto
  )
endif (WIN32)
```

After doing both of those changes, you'll need to run CMake "Configure" and "Generate" steps. The build with `mingw32-make` should succeed after that.