Security - Every Developer’s Responsibilty

Christoph Iserlohn

innoQ

About me

Senior Consultant @ innoQ

MacPorts Team member

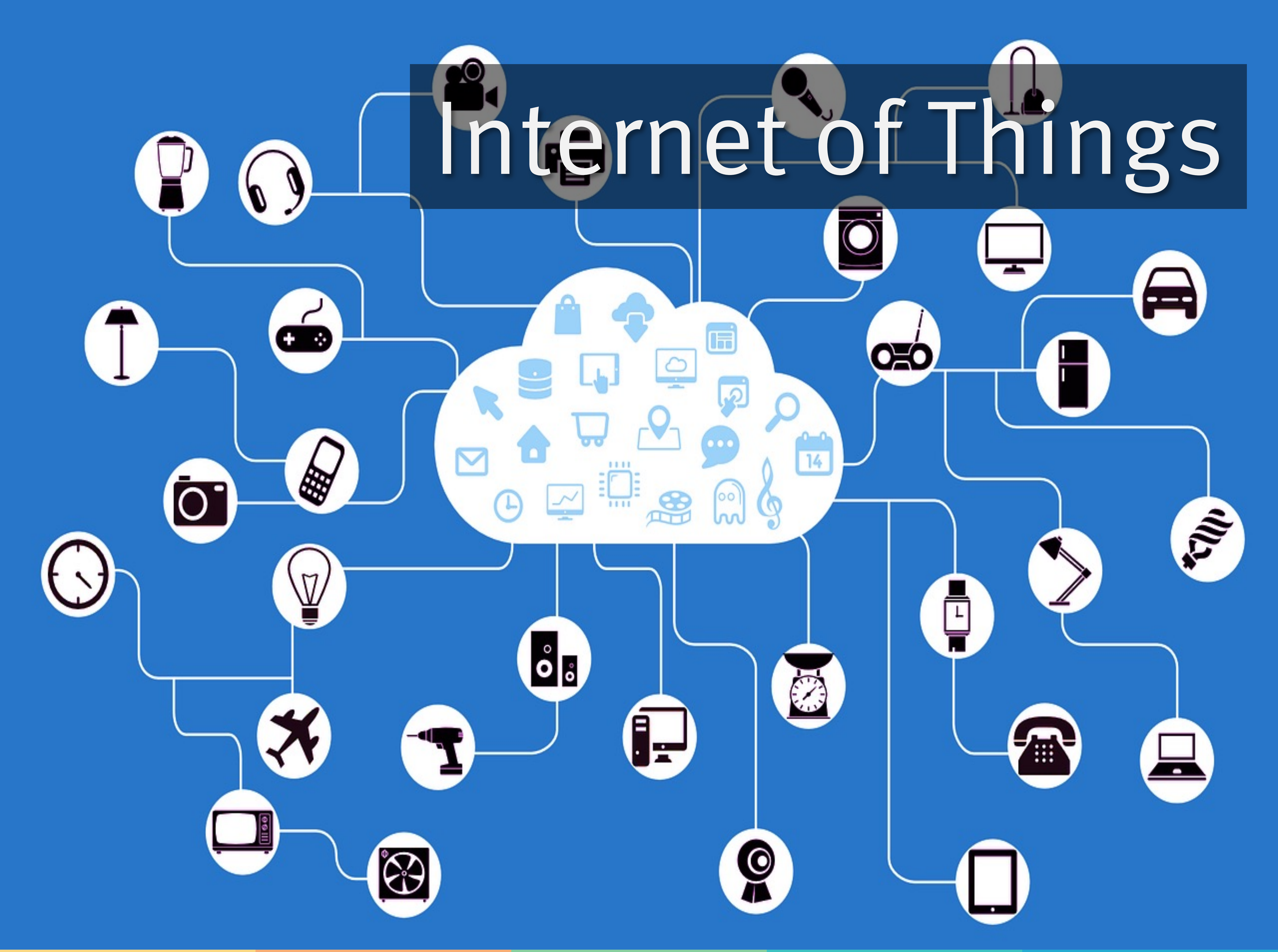
Agenda

Developers

Security

„Software is eating the world“

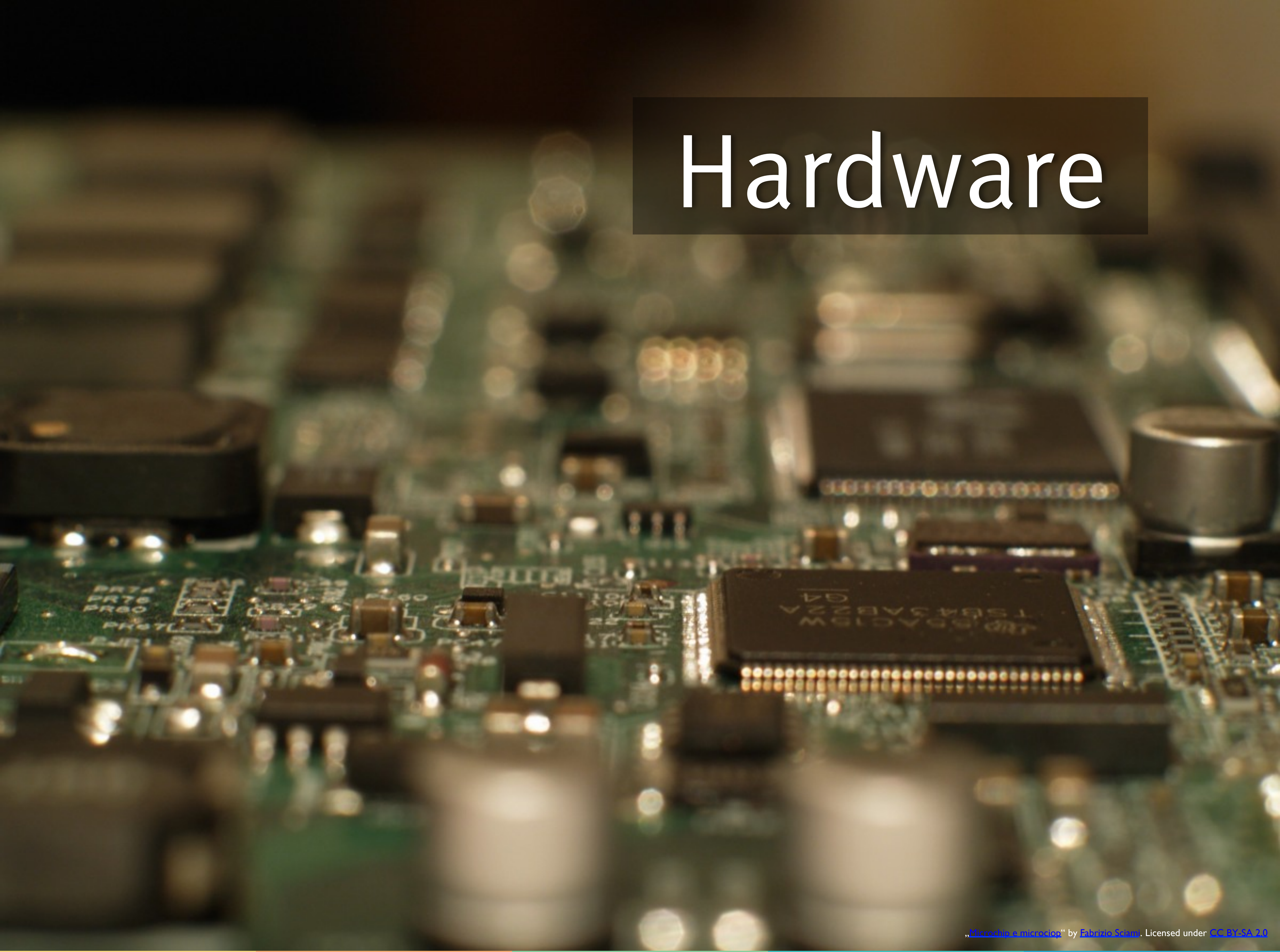
Internet of Things



The mess we‘re in

Security is not a product

Security dimesions



Hardware

Operating Systems

```
Current date is Tue  1-01-1980
Enter new date:
Current time is  7:48:27.13
Enter new time:


The IBM Personal Computer DOS
Version 1.10 (C)Copyright IBM Corp 1981, 1982

A>dir/w
COMMAND  COM    FORMAT   COM    CHKDSK   COM    SYS      COM    DISKCOPY COM
DISKCOMP COM    COMP     COM    EXE2BIN  EXE    MODE     COM    EDLIN    COM
DEBUG    COM    LINK     EXE    BASIC    COM    BASICA   COM    ART      BAS
SAMPLES  BAS    MORTGAGE BAS    COLORBAR BAS    CALENDAR BAS    MUSIC    BAS
DONKEY   BAS    CIRCLE   BAS    PIECHART BAS    SPACE    BAS    BALL     BAS
COMM     BAS
       26 File(s)
A>dir command.com
COMMAND  COM      4959   5-07-82  12:00p
        1 File(s)
A>
```

```
link: function(scope, element, attr, ngSwitchController) {
  var watchExpr = attr.ngSwitch || attr.on,
      selectedTranscludes = [],
      selectedElements = [],
      previousElements = [],
      selectedScopes = [];

  scope.$watch(watchExpr, function ngSwitchWatchAction(value) {
    var i, ii;
    for (i = 0, ii = previousElements.length; i < ii; ++i) {
      previousElements[i].remove();
    }
    previousElements.length = 0;

    for (i = 0, ii = selectedScopes.length; i < ii; ++i) {
      var selected = selectedElements[i];
      selectedScopes[i].$destroy();
      previousElements[i] = selected;
      $animate.leave(selected, function() {
        previousElements.splice(i, 1);
      });
    }

    selectedElements.length = 0;
    selectedScopes.length = 0;

    if ((selectedTranscludes = ngSwitchController.cases['!' + value] || ngSwitchC
      scope.$eval(attr.change);
      forEach(selectedTranscludes, function(selectedTransclude) {
        var selectedScope = scope.$new();
        selectedScopes.push(selec
```

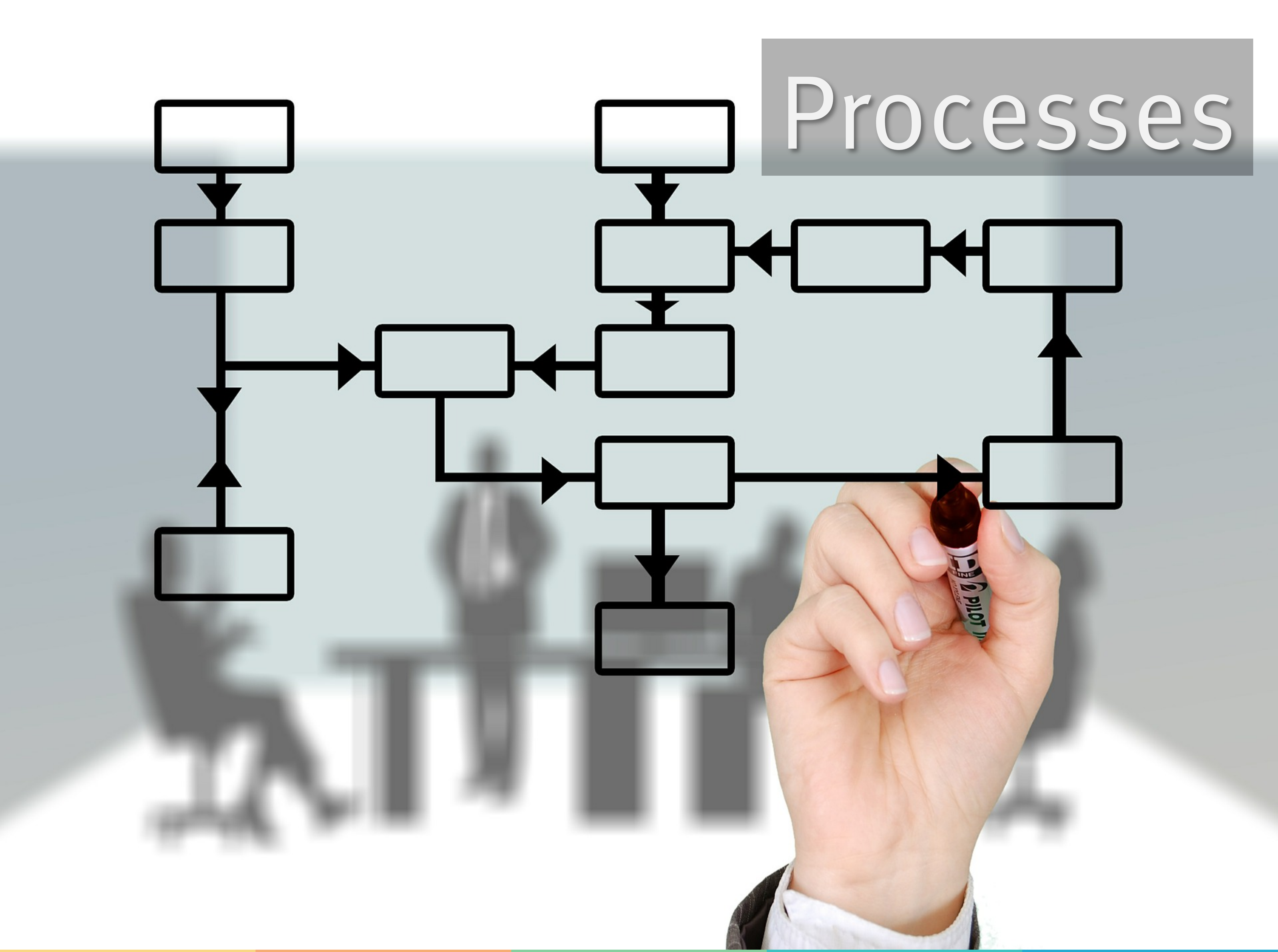
Software

Network

People

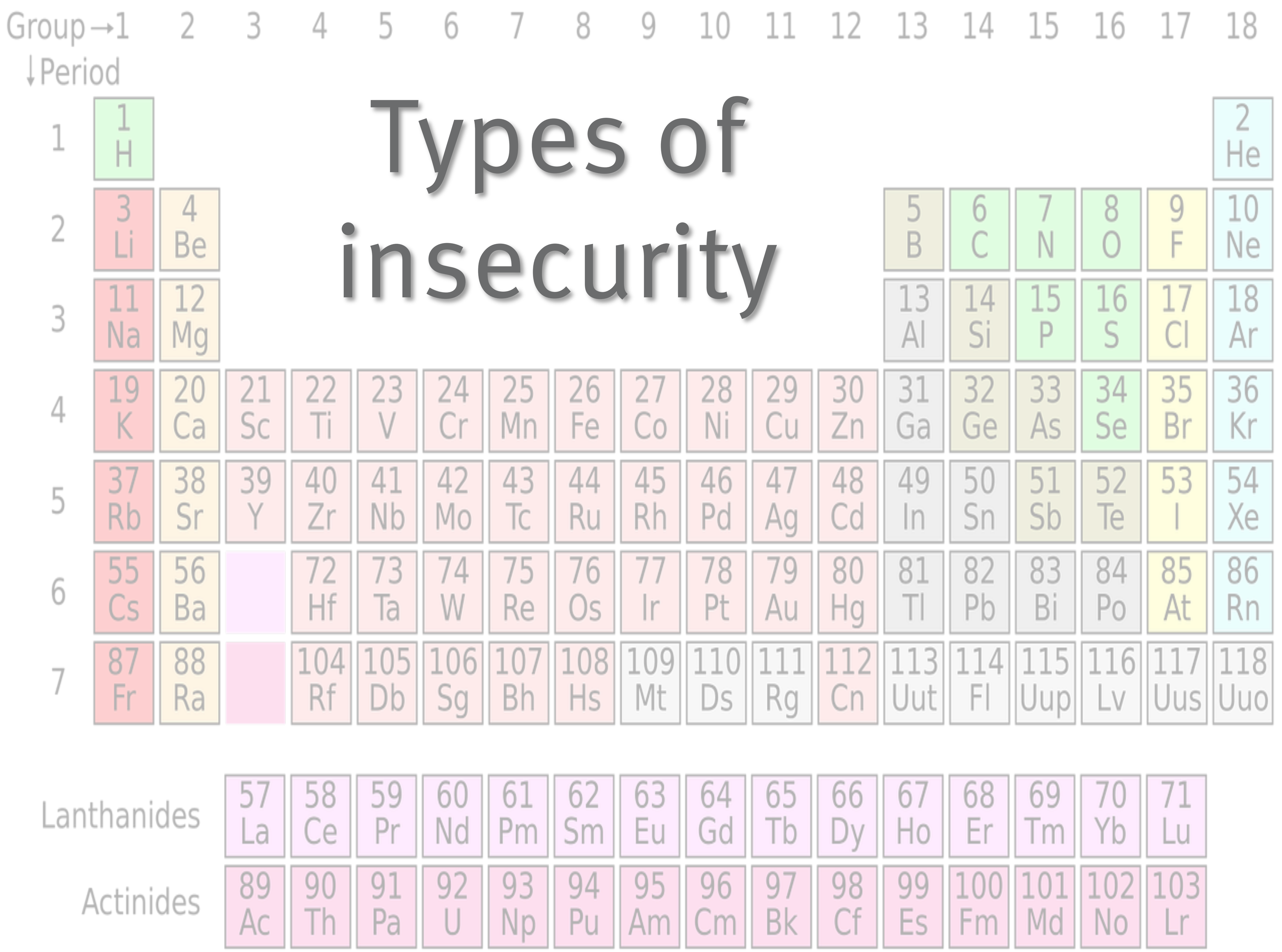
Organizations

Processes



Buildings

Law

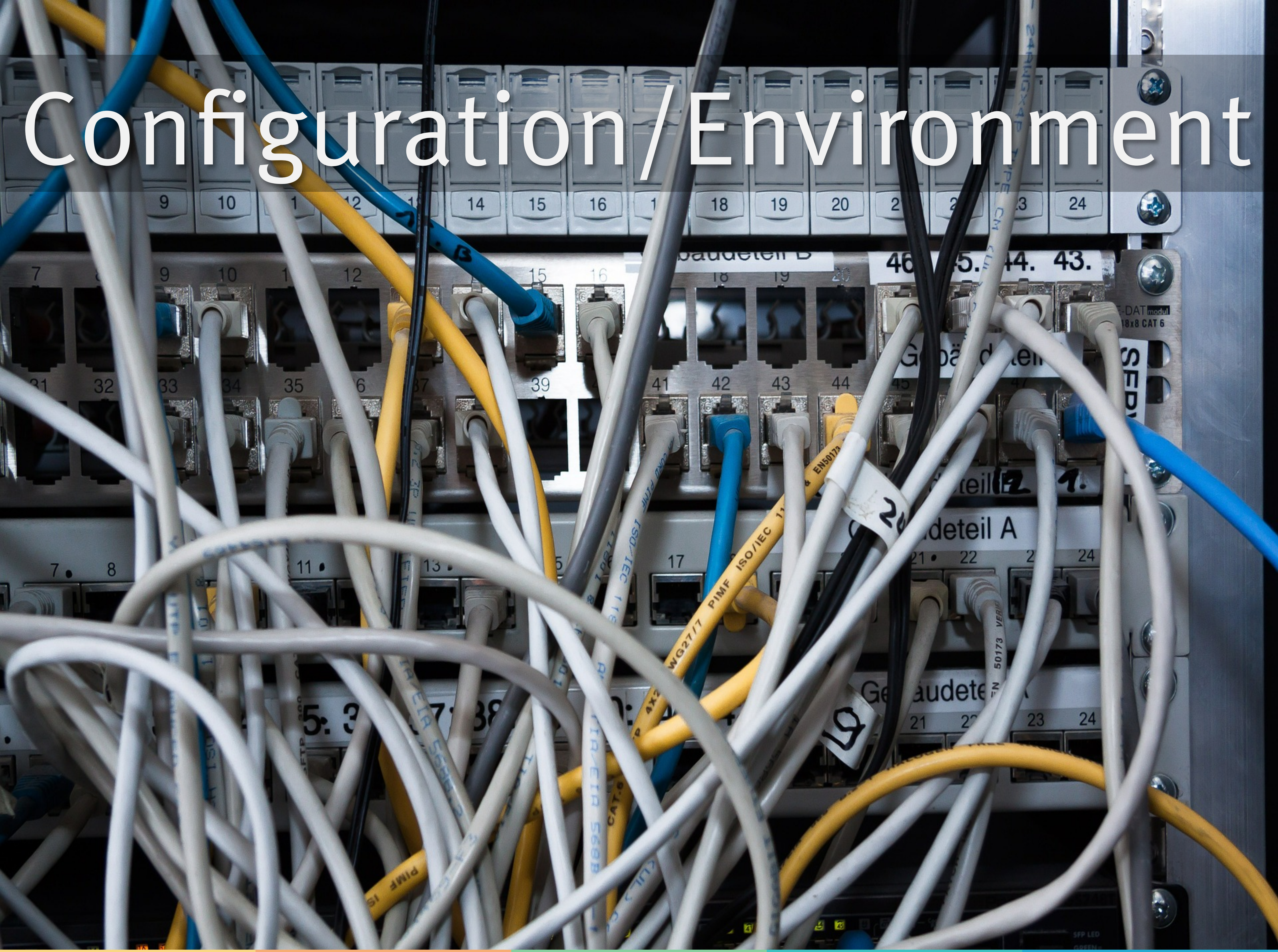


Types of insecurity

Trusting user input

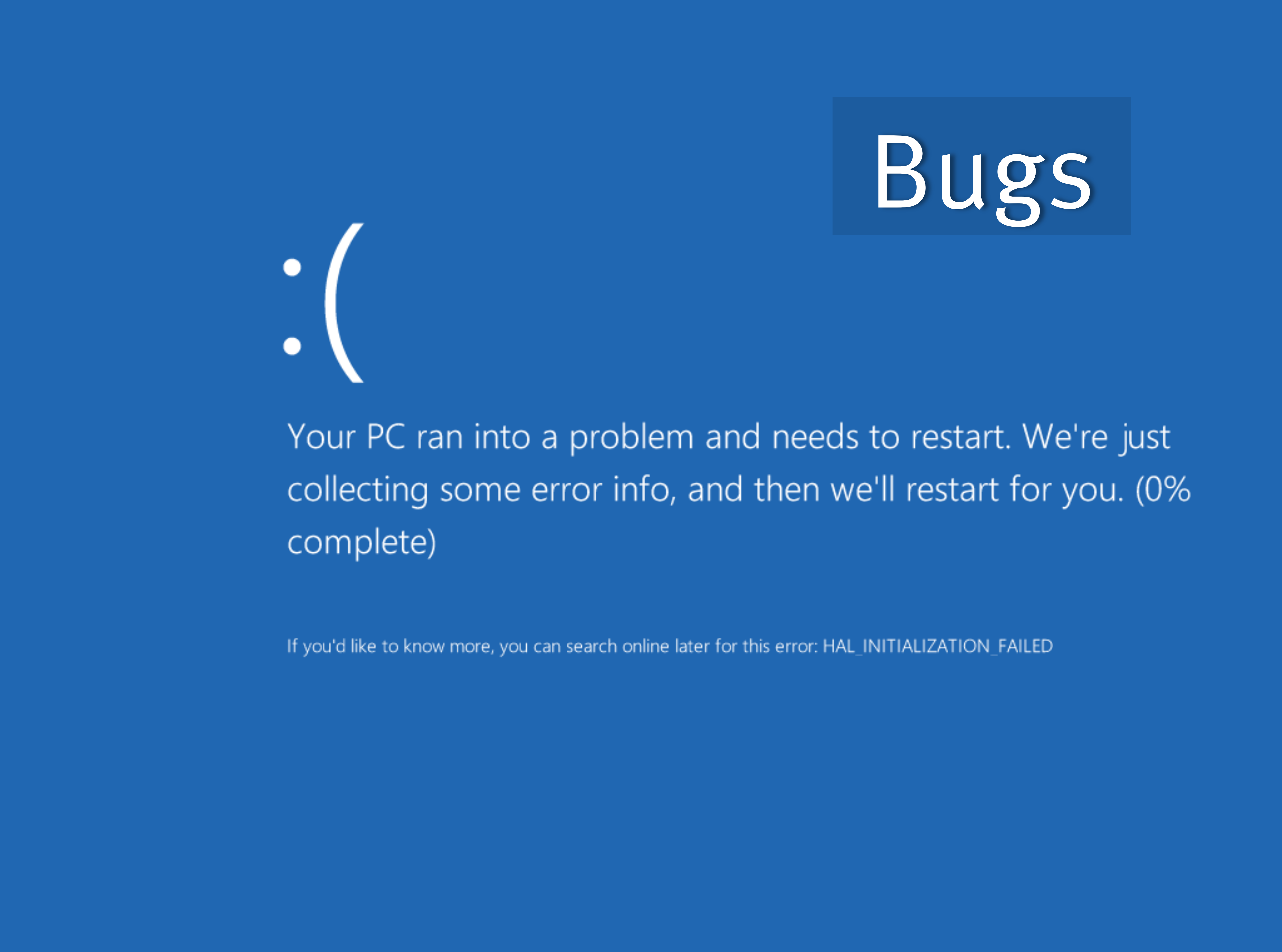
Logic errors /Design flaws

Configuration/Environment



Cryptographic weaknesses

Bugs


:(

Your PC ran into a problem and needs to restart. We're just collecting some error info, and then we'll restart for you. (0% complete)

If you'd like to know more, you can search online later for this error: HAL_INITIALIZATION_FAILED

Security should be

libsodium

```c
#include <sodium.h>

int main(void)
{
    if (sodium_init() == -1) { return 1; }

    const unsigned char message[] = "The quick brown fox jumps over the lazy dog";

    int message_len = sizeof message;
    int encrypted_len= message_len + crypto_secretbox_MACBYTES;

    unsigned char nonce[crypto_secretbox_NONCEBYTES];
    unsigned char key[crypto_secretbox_KEYBYTES];
    unsigned char encrypted[encrypted_len];
    unsigned char plain[message_len];

    randombytes_buf(nonce, sizeof nonce);
    randombytes_buf(key, sizeof key);

    crypto_secretbox_easy(encrypted, message, message_len, nonce, key);

    if (crypto_secretbox_open_easy(plain, encrypted, encrypted_len, nonce, key) != 0) {
        printf("Message has been forged!");
        return 1;
    }

    printf("Message to encrypt: %s\nCiphertext: ", message);

    for(int i = 0; i < ciphertext_len; i++) {
        printf("%02x", ciphertext[i]);
    }

    printf("\nDecrypted message: %s\n", decrypted);
}
```

libcrypto

```c
#include <openssl/conf.h>
#include <openssl/evp.h>
#include <openssl/err.h>
#include <string.h>

int main (void) {
  unsigned char *key = (unsigned char *)"01234567890123456789012345678901";
  unsigned char *iv = (unsigned char *)"0123456789012345";
  unsigned char *plain = (unsigned char *)"The quick brown fox jumps over the lazy dog";

  unsigned char ciphertext[128];
  unsigned char decryptedtext[128];

  int decryptedtext_len, ciphertext_len;

  ERR_load_crypto_strings();
  OpenSSL_add_all_algorithms();
  OPENSSL_config(NULL);

  ciphertext_len = encrypt(plain, strlen ((char *)plain), key, iv, ciphertext);

  printf("Ciphertext is:\n");
  BIO_dump_fp (stdout, (const char *)ciphertext, ciphertext_len);

  decryptedtext_len = decrypt(ciphertext, ciphertext_len, key, iv, decryptedtext);

  decryptedtext[decryptedtext_len] = '\0';

  printf("Decrypted text is:\n");
  printf("%s\n", decryptedtext);

  EVP_cleanup();
  ERR_free_strings();
}
```

libcrypto - continued

```c
int encrypt(unsigned char *plaintext, int plaintext_len, unsigned char *key,
  unsigned char *iv, unsigned char *ciphertext)
{
  EVP_CIPHER_CTX *ctx;

  int len;

  int ciphertext_len;

  if(!(ctx = EVP_CIPHER_CTX_new())) handleErrors();

  if(1 != EVP_EncryptInit_ex(ctx, EVP_aes_256_cbc(), NULL, key, iv))
    handleErrors();

  if(1 != EVP_EncryptUpdate(ctx, ciphertext, &len, plaintext, plaintext_len))
    handleErrors();
  ciphertext_len = len;

  if(1 != EVP_EncryptFinal_ex(ctx, ciphertext + len, &len)) handleErrors();
  ciphertext_len += len;

  EVP_CIPHER_CTX_free(ctx);

  return ciphertext_len;
}
```

libcrypto - continued

```c
int decrypt(unsigned char *ciphertext, int ciphertext_len, unsigned char *key,
  unsigned char *iv, unsigned char *plaintext)
{
  EVP_CIPHER_CTX *ctx;

  int len;

  int plaintext_len;

  if(!(ctx = EVP_CIPHER_CTX_new())) handleErrors();

  if(1 != EVP_DecryptInit_ex(ctx, EVP_aes_256_cbc(), NULL, key, iv))
    handleErrors();

  if(1 != EVP_DecryptUpdate(ctx, plaintext, &len, ciphertext, ciphertext_len))
    handleErrors();
  plaintext_len = len;

  if(1 != EVP_DecryptFinal_ex(ctx, plaintext + len, &len)) handleErrors();
  plaintext_len += len;


  EVP_CIPHER_CTX_free(ctx);

  return plaintext_len;
}
```

libcrypto - continued

```
void handleErrors(void)
{
  ERR_print_errors_fp(stderr);
  abort();
}
```

stackoverflow

```java
// Create a trust manager that does not validate certificate chains
TrustManager[] trustAllCerts = new TrustManager[] { new X509TrustManager() {
    public java.security.cert.X509Certificate[] getAcceptedIssuers() {
        return null;
    }
    public void checkClientTrusted(X509Certificate[] certs, String authType) {}
    public void checkServerTrusted(X509Certificate[] certs, String authType) {}
    }};
// Install the all-trusting trust manager
final SSLContext sc = SSLContext.getInstance("SSL");
sc.init(null, trustAllCerts, new java.security.SecureRandom());
HttpsURLConnection.setDefaultSSLSocketFactory(sc.getSocketFactory());
// Create all-trusting host name verifier
HostnameVerifier allHostsValid = new HostnameVerifier() {
    public boolean verify(String hostname, SSLSession session) {
        return true;
    }
    };

// Install the all-trusting host verifier
HttpsURLConnection.setDefaultHostnameVerifier(allHostsValid);
```

You can‘t be a professional…

...without knowing the basics.

Thank you!

- Questions ?
- Comments ?

Christoph Iserlohn

christoph.iserlohn@innoq.com