MeaWallet

C++ Developer’s Home Task

Version: 1.0

General description

Create an audit log building program and corresponding audit log parsing program. Audit logs are stored encrypted and the resulting ciphertext is protected with MAC (Message Authentication Code). Keys can be hard-coded and don't require protection for this task. Key length should be 16-bytes. Encryption should be performed with AES-CBC encryption algorithm, random IV (Initialization Vector). MAC should also be calculated with AES-CBC MAC or HMAC algorithm. If HMAC algorithm is used then key length for MAC calculation can be longer. For encryption/decryption and MAC calculation/verification purposes any cryptography library can be used. Code has to be written in C++ (C++11, C++14, C++17 features can be used). Preferred execution environment is 64-bit Linux but can also be Windows or Mac OS X.

Audit log builder

Builder will store multiple sample audit log entries in the `audit_log` file. Below 3 sample records are shown:

```
{
      created_at: 2020-12-01 12:00:00
      user_id: Employee_1
      event_type: 3
      additional_data: "User logged in."
      error_codes: "678|159|262"
   },
   {
      created_at: 2020-12-01 12:01:00
      user_id: Employee_1
      event_type: 1
      additional_data: "User logged out."
      error_codes: null
   },
   {
      created_at: 2020-12-01 12:05:00
      user_id: null
      event_type: 5
      additional_data: "App data wiped."
```

```
    error_codes: null
}
...
```

Format of entries is not important (can be Json, but also can be binary format or other) but it should hold information as illustrated above (5 fields). Each entry is stored encrypted using the `audit_encryption` key and each resulting ciphertext is protected with a MAC using `audit_mac` key. It shouldn't be possible to see audit log file contents in cleartext or modify/delete audit log file. To prevent cases where records are added/deleted from `audit_log` file or entire `audit_log` file is deleted, create another file `secure_storage` and store the number of audit log records present in audit_log file. `secure_storage` file contents should also be protected with AES-CBC encryption (`storage_encryption` key) and AES-CBC MAC or HMAC (`storage_mac` key). Assume that it is not possible to delete the secure_storage file or remove its contents.

Audit log parser

Parser should open a `secure_storage` file and verify MAC then decrypt audit log record count from it if MAC matches. Afterwards the parser verifies the MAC of each record inside the `audit_log` file. If all MACs inside the `audit_log` file are valid then each record should be decrypted. Check if each record is consecutive based on creation timestamp (`created_at` field). Also compare if record count in `audit_log` matches record count retrieved from `secure_storage` file. If all checks are passed then each audit log can be printed on screen - format is not important but entries should at least be separated visually (for example, with newline character).

In case of questions, contact karlis.balcers@meawallet.com

~ DONE ~