PV204: 7-Zip JavaCard security

E Team
Agusti Bau[454255] , Aobakwe Mmokwa[448382] , Lubo Obratil[410282]

April 15, 2016

1 Introduction

7-Zip is open-source project (`http://www.7-zip.org/`) distributed under GNU LGPL, that allows user to compress and decompress files and optionally protect them with password. In normal work-flow, if user wishes to work with password protected file, he enters plain password into application and given file is then processed. This approach can be exploited in various ways.

Possible exploits

- Users don't use secure passwords very often ⇒ 6-7 character passwords can be brute-forced.
- User's computer may be infected with keylogger ⇒ attacker can obtain plain password.
- Password is directly used by application ⇒ can be retrieved from memory dump.

2 Proposed approach

Our goal would be to implement JavaCard applet which would serve as a password provider for 7-Zip application. Applet would be capable of persistent storage of master key, derivation of encryption and decryption keys to be used by 7-Zip and establishing of secure channel between token and application.

Application work-flow

1. User chooses file he wishes to compress and encrypt or decrypt.
2. 7-Zip establishes secure channel with JavaCard token.
3. User provides his PIN that is then checked against PIN stored in token.
4. 7-Zip will encrypt or decrypt file.
   (a) **Encryption**
      i. 7-Zip will request encryption key.
      ii. Token will increment its inner counter and based on this counter and master key will derive encryption key.
      iii. Token will export the encryption key into application.
      iv. 7Zip will request token's current counter and save it into unencrypted part of archive's metadata.
      v. File is compressed and encrypted by 7-Zip.
   (b) **Decryption**
      i. 7-Zip will retrieve counter from archive's metadata.
      ii. 7-Zip will send counter to token.
      iii. Based on received counter and master key, token will derive decryption key.
      iv. Decryption key is used for archive decryption by 7-Zip.
      v. Archive is decompressed.
5. Operation is finished, connection to token is closed.

Increased security of proposed approach

- Token uses secure channel ⇒ prevented eavesdropping on channel between token and application.
- Only PIN is entered by user ⇒ prevents key retrieval using key-logger, since PIN alone is useless without token.
- Each new archive is protected by different key ⇒ key extracted from memory can be used on only one file.
- Password created by token ⇒ prevents brute-force and dictionary attacks, generated password can fully utilize AES256 encryption which is used by 7-Zip.
- User or admin PIN is needed to use token ⇒ attacker can't just steal token.

Scenarios this approach won't prevent

- Attacker will steal token with PIN written on it.
- Attacker will steal token and will pay rogue admin to set user PIN to `0000`.
- Attacker will steal token and will torture user until he or she tells him the PIN.

Additional specification of JavaCard token

- Master key can be generated and erased on the token, but can't be extracted by anyone.
- All key operations are available only after successful user PIN verification.
- Device is blocked after 3 unsuccessful user PIN tries. It can be unlocked by admin PIN.
- User PIN can be set by user or admin.
- Admin PIN can be set only by admin.
- Admin PIN is 256 bit long key which can not be brute-forced, therefore device can't be blocked by admin PIN tries.