Security Engineering - Winter 2017

Sambuddho Chakravarty

April 16, 2017

Assignment 4 (Total points: 50)

Due date: April 30. Time: 23:59 Hrs.

File Authentication and Integrity Protection using HMAC

This assignment expands on the previous assignment by adding HMAC authentication to files. In addition to encrypting the file you also need to generate a HMAC signature of the file using key that was used to encrypt the contents. You need to additional two additional – `fauth_gen_hmac` and `fauth_test_auth`. The first program, `fauth_gen_hmac` generates a HMAC signature for the file and stores it along with the file. The HMAC key is the same as the key used to encrypt the file. The second program `fauth_test_auth` recreates the signature of the file contents and matches it against the stored HMAC (the one generated with `fauth_gen_hmac`).

To demonstrate the functionality of the programs, you would require to make some change to the file contents (directly WITHOUT using the previously written functions, as they may block access based on ACL rules), WITHOUT changing the original HMAC signature. Thereafter, if you run the program `fauth_test_auth`, it must fail, throwing an appropriate error message.

Feel free to consider other possible assumptions. DO NOT forget to list the assumptions in the system description that you would submit.

Grading Rubric

- Successful compilation using Makefile – 5 points.
- Use of OpenSSL library functions for encryption, decryption, HMAC signatures *etc.* for authenticating users and performing encryption and decryption operations – 20 points.
- Correct functionality of the programs – 20 points.
- Write-up describing the functionality/assumptions/how the various programs work *etc.* – 5 points.

Deadlines

- Submission made on or before April 30, 2017 (23:59 hrs) – No points deducted.
- Submissions made on or after April 30, 2017 but on or before May 1, 2017 (23:59 hrs) – 5 points deducted.
- Submitted after May 1, 2017 – no grade.