Security Engineering - Winter 2017

Sambuddho Chakravarty

April 1, 2017

Assignment 3 (Total points: 50)

Due date: April 15. Time: 23:59 Hrs.

File Encryption

The objective of this assignment is to familiarize you with using OpenSSL library routines. You would need to start with the set-up you used in assignment 1. To this you need to add some new programs like 'fget_decrypt', 'fput_encrypt'. The 'fput_encrypt' creates a file, using text entered through standard input while storing in encrypted in the file name specified as a program argument. The file contents need to be encrypted using symmetric encryption such as AES. Upon executing the command 'fput_encrypt', it should prompt the user for a passphrase that is to be used to generate the salt, IV and key to encrypt the file. The file should then be encrypted using the generated salt, IV and key. Similarly, the 'fput_decrypt' program should prompt the users for the passphrase from the user to decrypt the file and should print it out for the user. Again, these programs should be setuid programs which could access the users' files. These programs should also switch to the effect users' ID using seteuid() (or related) systems call(s).

The files now would additionally have a access control bit – 'd' , which stands for accessing (reading from or writing to files that are encrypted).

Just like the usual ACL bits, the files would additionally have this bit which would signify if the files can be read (after being decrypted) or written to (decrypted, edited and eventually written back to the disk.

For each user, you also need to store a file in simple_slash/etc/shadow, which should store the hashed user passwords. Every time a user tries to access an encrypted file, the system checks the ACL permissions associated with the file to determine if the user is allowed to encrypt and decrypt the file. If the user is allowed, the system prompts the user for the password which is hashed and validated with the password (NOTE: password needs to be checked ONLY for the owner). If the passwords match, then the system decrypts the files using the key (generated by hashing the password) along with the IV (stored with the file). Ideally you would want to stored hashed passwords. You could use these as the keys also.

You need to check for every corner case with regards to the functionality. 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, hashing *etc.* for authenticating users and performing encryption and decryption operations – 20 points.
- Correct functionality of the programs along with correctly functioning ACLs for handling the 'd' bit for decryption and encryption – 20 points.
- Write-up describing the functionality/assumptions/how the various programs work *etc.* – 5 points.

Deadlines

- Submission made on or before April 15, 2017 (23:59 hrs) – No points deducted.
- Submissions made on or after April 15, 2017 but on or before April 17, 2017 (23:59 hrs) – 5 points deducted.
- Submitted after April 17, 2017 – no grade.