Cleartext storage of sensitive information in file

Name: Cleartext storage of sensitive information in file

Description: Storing sensitive information in cleartext can expose it to an attacker.

ID: cpp/cleartext-storage-file

Kind: problem

Severity: warning

Precision: medium

Sensitive information that is stored unencrypted is accessible to an attacker who gains access to the storage.

Recommendation

Ensure that sensitive information is always encrypted before being stored, especially before writing to a file. It may be wise to encrypt information before it is put into a buffer that may be readable in memory.

In general, decrypt sensitive information only at the point where it is necessary for it to be used in cleartext.

Example

The following example shows two ways of storing user credentials in a file. In the 'BAD' case, the credentials are simply stored in cleartext. In the 'GOOD' case, the credentials are encrypted before storing them.

```
1  void writeCredentials() {
2    char *password = "cleartext password";
3    FILE* file = fopen("credentials.txt", "w");
4
5    // BAD: write password to disk in cleartext
6    fputs(password, file);
7
8    // GOOD: encrypt password first
9    char *encrypted = encrypt(password);
10   fputs(encrypted, file);
11 }
```

References

- M. Dowd, J. McDonald and J. Schuhm, *The Art of Software Security Assessment*, 1st Edition, Chapter 2 - 'Common Vulnerabilities of Encryption', p. 43. Addison Wesley, 2006.
- M. Howard and D. LeBlanc, *Writing Secure Code*, 2nd Edition, Chapter 9 - 'Protecting Secret Data', p. 299. Microsoft, 2002.
- Common Weakness Enumeration: CWE-313.