Use of a broken or risky cryptographic algorithm

Name: Use of a broken or risky cryptographic algorithm

Description: Using broken or weak cryptographic algorithms can allow an attacker to compromise security.

ID: cpp/weak-cryptographic-algorithm

Kind: problem

Severity: error

Precision: medium

Using broken or weak cryptographic algorithms can leave data vulnerable to being decrypted.

Many cryptographic algorithms provided by cryptography libraries are known to be weak, or flawed. Using such an algorithm means that an attacker may be able to easily decrypt the encrypted data.

Recommendation

Ensure that you use a strong, modern cryptographic algorithm. Use at least AES-128 or RSA-2048.

Example

The following code shows an example of using the `advapi` windows API to decrypt some data. When creating a key, you must specify which algorithm to use. The first example uses DES which is an older algorithm that is now considered weak. The second example uses AES, which is a strong modern algorithm.

```
1  void advapi() {
2    HCRYPTPROV hCryptProv;
3    HCRYPTKEY hKey;
4    HCRYPTHASH hHash;
5    // other preparation goes here
6
7    // BAD: use 3DES for key
8    CryptDeriveKey(hCryptProv, CALG_3DES, hHash, 0, &hKey);
9
10   // GOOD: use AES
11   CryptDeriveKey(hCryptProv, CALG_AES_256, hHash, 0, &hKey);
12 }
```

References

- NIST, FIPS 140 Annex a: Approved Security Functions.
- NIST, SP 800-131A: Transitions: Recommendation for Transitioning the Use of Cryptographic Algorithms and Key Lengths.
- Common Weakness Enumeration: CWE-327.