- deallocare la memoria il più presto possibile (distruzione degli oggetti) laddove tale operazione non pregiudichi la sicurezza dell'applicazione;
- compilare il software per la piattaforma di utilizzo (es: non compilare per architettura hardware 64-bit se non è necessario).

5.2.2 Password nel codice sorgente

I dati di accesso (username/password/nome db/ecc..) ai database o a sistemi di altra natura non devono mai essere inseriti all'interno dei sorgenti.
Nei casi in cui non sia possibile, tali dati devono apparire in forma cifrata. Per le chiavi di cifratura e in generale per tutte le informazioni riservate valgono le stesse indicazioni.

5.2.3 Privilegi esecutivi minimi

Quando l'applicazione viene avviata all'interno del sistema operativo, porta con sé i privilegi dell'utenza che effettua l'operazione. L'applicazione non deve essere lanciata con i privilegi amministrativi.

5.2.4 Metodi TRACE e TRACK

Uno dei principi di sicurezza più saggi afferma che ciò che non viene utilizzato dovrebbe essere disabilitato. Nelle applicazioni Web è obbligatoria la disattivazione lato server del metodo HTTP TRACE o del metodo TRACK (utilizzato in ambienti Microsoft IIS). Tali metodi consentono al client di vedere ciò che viene ricevuto dal web server. Tali informazioni possono poi essere utilizzate per organizzare un attacco di Cross Site Scripting. Si parla di "Cross Site Tracing" (XST).

5.2.5 Assenza di codice malevolo

L'applicazione non deve contentere alcun tipo malware (malicious software): virus, trojan, rootkit, worms, ramsonware, ecc.
Sono da considerare potenzialmente pericolose anche le backdoor amministrative, poiché consentono l'accesso alle macchine in rete bypassando il processo di autenticazione. Un attaccante che trovasse il modo di manomettere una backdoor amministrativa, potrebbe penetrare nelle macchine e prenderne il controllo.

5.2.6 Fattore integrità

Il concetto di integrità del software include la resilienza agli attacchi informatici e alle violazioni della privacy, ma essenzialmente sta a indicare che possano essere impedite modifiche non autorizzate.
La fase di progettazione e la successiva fase d'implementazione devono assicurare che tutti gli errori e le eccezioni rilevati durante il processamento e l'elaborazione dei dati acquisiti in ingresso siano correttamente gestiti, in modo che non causino il danneggiamento o la perdita di integrità delle informazioni.

5.2.7 Input character validation

L'applicazione deve assicurare, attraverso opportuni meccanismi di convalida, che tutti i parametri in input, specificati dall'utente, siano congruenti a quanto atteso.
In particolare, sui dati acquisiti in ingresso, l'applicazione deve prevedere l'implementazione di meccanismi di controllo che limitino il set di caratteri o valori, inseribili dall'utente, solo a quelli congruenti ai campi richiesti e/o alle form di pertinenza, al fine di identificare e annullare gli effetti dei seguenti errori:

- Valori out-of-range o non pertinenti (ad esempio l'immissione di caratteri non numerici nel campo "anno di nascita");
- Caratteri invalidi negli stream o nei data field;
- Dati mancanti o incompleti;
- Limite del minimo volume di dati richiesti non soddisfatto o del massimo volume di dati acquisibile in ingresso raggiunto;