- **Tentativi di accesso a risorse inibite**: si devono tracciare tutti i tentativi di accesso a risorse inibite ai servizi come, ad esempio, tentativi di accesso alla root di un server web, modifica a configurazioni per mezzo di credenziali non appropriate, etc. (Cfr. [paragrafo 5.6]);
- **Violazioni delle policy configurate**: si devono tracciare le violazioni o i tentativi di bypass delle regole di autorizzazione che definiscono ruolo e permessi assegnati all'utente nonché le operazioni ad esso concesse in base alla tipologia di profilatura (Cfr. [paragrafo 5.6]);
- **Process Issue**: si devono tracciare gli avvisi, generati in ambito Server Applicativo, relativi all'esecuzione di moduli applicativi che risultano diversi in quantità e dimensione rispetto a quanto atteso/definito in fase di progettazione/realizzazione dell'applicativo stesso (ad es. numero eccessivo di istanze duplicate, esecuzione di istruzioni non previste, eccessiva occupazione di memoria, etc.) -(Cfr. [paragrafo 6.1.7]);
- **Funzioni input/output anomale**: si devono tracciare i tentativi inaspettati di dichiarazioni di funzioni e/o comandi in input ed in output (Cfr. [paragrafo 5.2.7 , 5.2.8, cap. 6]);
- **Disattivazione anomala del meccanismo di tracciamento**: devono essere osservati e tracciati tutti i cambiamenti di stato (attivo ↔ disattivo) delle funzioni di tracciamento e generazione allarmi, su tutte le componenti funzionalmente coinvolte. Altresì, è necessario tenere sempre sotto controllo le attività di download/upload dell'utente, al quale è stato consentito l'accesso al sistema, al fine di individuare situazioni anomale (generazione di allarmi laddove la quantità di dati superi una certa soglia che tiene conto del livello/ruolo di accesso dell'utente).

5.5 Compilazione dell'applicazione

Per la compilazione del codice dell'applicazione si raccomanda l'adozione dei criteri riportati nei paragrafi (Cfr. [5.5.1,5.5.2]) che seguono.

5.5.1 Stack Canary

I sorgenti dell'applicazione e delle librerie che la compongono (DLL o altre forme comparabili in ambienti operativi differenti) devono essere compilati con funzionalità di stack canary. A runtime viene impostato un valore (spesso un intero) in memoria e viene verificato che non venga sovrascritto da un eventuale buffer overflow, dopo una chiamata allo stack. Ciò permette di bloccare gli effetti di un buffer overflow in tempo utile. In fase di compilazione, devono essere attivate opzioni di anti-sovversione dei puntatori ai gestori delle eccezioni (ad esempio SafeSEH), relativamente alla piattaforma dell'applicazione.

5.5.2 Correttezza del sorgente

La compilazione dei sorgenti deve terminare senza alcun tipo di warning.

5.6 Ambiente operativo dell'applicazione

In merito agli ambienti operativi di sviluppo e test delle applicazioni, si raccomanda l'adozione dei criteri riportati nei paragrafi (Cfr. [5.6.1 - 5.6.6]) che seguono.

5.6.1 Separazione degli ambienti

I sistemi di sviluppo, test e produzione devono essere separati fisicamente e/o logicamente.

5.6.2 Test dell'Applicazione

- L'applicazione deve essere consegnata e portata in produzione/esercizio solo dopo essere stata verificata la rispondenza ai requisiti dati.
- I casi di test devono includere controlli sull'usabilità dell'applicazione, sulla sicurezza e sulla compatibilità con l'infrastruttura hardware/software in cui andrà installata.
- È raccomandato l'utilizzo di appositi strumenti di stress test prima dell'avvio in esercizio dell'applicazione, al fine di certificare la corretta implementazione delle procedure di input data validation e security menzionate in questo documento.