- External reporting;
- Porte TCP/UDP (network socket).

#### *6.1.3.4 Punti di Uscita (Exit Points)*

È importante identificare da dove l'applicazione invia i dati all'utente o ai sistemi esterni, dando priorità ai punti di uscita in cui l'applicazione scrive dati che includono l'input proveniente dall'utente o dati provenienti da fonti non attendibili, ad esempio basi dati condivise.
A titolo esplicativo si elencano di seguito ulteriori esempi di Exit Points:

- File di Log;
- Database;
- Interfacce esterne (es. web service).

### 6.1.4 Identificazione delle minacce

In questa fase, è possibile individuare minacce e attacchi che potrebbero compromettere l'applicazione e gli obiettivi di sicurezza. Il processo di identificazione consiste in sessioni di brainstorming tra i team di sviluppo e test. Idealmente, il team di lavoro è costituito da architetti software, professionisti della sicurezza, sviluppatori, tester e amministratori di sistema.
Esistono due approcci per affrontare questa fase:

- Iniziare, elencando minacce e attacchi comuni. Con questo approccio, si inizia con un elenco di minacce comuni raggruppate per categorie di vulnerabilità. Successivamente, occorre adattare tale elenco alla propria architettura. Ad esempio, utilizzando gli scenari identificati per esaminare i flussi di dati, prestando particolare attenzione ai punti di ingresso e in particolare a quelli che attraversano i confini di fiducia. In questo modo si potranno eliminare immediatamente alcune minacce, in quanto non applicabili ai casi d'uso.
- Utilizzare un approccio basato su domande. Un approccio basato su questionari può aiutare a identificare le minacce e i possibili attacchi. La categorizzazione STRIDE si basa su categorie di minacce molto estese, quali spoofing (assunzione impropria di identità), manomissione di dati, ripudio, divulgazione indesiderata di informazioni e interruzione di servizio. Il modello STRIDE deve essere usato per porre domande relative a qualsiasi aspetto dell'architettura e del design dell'applicazione. Questo è un approccio basato su obiettivi, in cui si prendono in considerazione tutti gli obiettivi di un possibile aggressore (punto di vista di un attaccante).

Per identificare le minacce, si esaminano tutti i livelli dell'applicazione, livello per livello e funzione per funzione. Ponendo l'attenzione sulle categorie di vulnerabilità, ci si concentra sulle aree in cui vengono spesso effettuati errori di sicurezza. Occorre identificare le potenziali minacce e le possibili azioni che un aggressore potrebbe tentare di eseguire per sfruttare le vulnerabilità a cui l'applicazione è esposta. Durante questa attività di identificazione delle minacce si eseguono le seguenti attività:

- Identificazione delle minacce e degli attacchi comuni.
- Identificazione delle minacce annidate nei casi d'uso.
- Identificazione delle minacce annidate nei flussi di dati.

#### *6.1.4.1 Identificazione delle minacce e attacchi comuni*

Esistono una serie di minacce e attacchi comuni che si basano su vulnerabilità di carattere comune. Questa sezione identifica una serie di domande chiave da porsi per ciascuna categoria.
Autenticazione:

- Come potrebbe un aggressore rubare una identità?
- Come potrebbe un utente malintenzionato accedere all'archivio delle credenziali?
- Come potrebbe un aggressore portare un attacco? Come vengono memorizzate le credenziali dell'utente e quali criteri di codici di accesso vengono applicati?

- Come può un utente malintenzionato modificare, intercettare o eludere il meccanismo di ripristino delle credenziali dell'utente?

Autorizzazione:

- Come potrebbe un utente malintenzionato influenzare i controlli di autorizzazione per accedere a operazioni privilegiate?
- Come potrebbe un utente malintenzionato elevare i propri privilegi?

Input e dati di convalida:

- Come potrebbe un utente malintenzionato iniettare comandi SQL?
- Come potrebbe un utente malintenzionato eseguire un attacco di cross-site scripting?
- Come potrebbe un utente malintenzionato eludere la validazione degli input?
- Come potrebbe un utente malintenzionato inviare un input non valido per influenzare la logica di protezione adottata sul server?
- Come potrebbe un utente malintenzionato sollevare un errore di input per bloccare l'applicazione?

Gestione della configurazione:

- Come potrebbe un utente malintenzionato accedere alle funzioni di amministratore della configurazione?
- Come potrebbe un utente malintenzionato accedere ai dati di configurazione dell'applicazione?

Dati sensibili:

- Dove e come l'applicazione memorizza i dati sensibili?
- Quando e in quale punto i dati sensibili vengono passati attraverso la rete?
- Come potrebbe un utente malintenzionato visualizzare i dati sensibili?
- Come potrebbe un utente malintenzionato manipolare i dati sensibili?

Gestione delle sessioni:

- Si utilizza un algoritmo di crittografia personalizzato e ci si fida di tale algoritmo?
- Come potrebbe un aggressore prendere il controllo della sessione di un utente?
- Come potrebbe un utente malintenzionato visualizzare o modificare lo stato della sessione di un altro utente?

Crittografia:

- Di cosa ha bisogno un aggressore per sovvertire il meccanismo di crittografia adottato?
- Come potrebbe un utente malintenzionato ottenere l'accesso alle chiavi crittografiche?
- Quali standard crittografici si stanno utilizzando? Quali sono gli attacchi noti su tali standard?
- Si vuole adottare un proprio meccanismo di crittografia?
- In che modo la tipologia di distribuzione potenzialmente influenzerà la scelta dei metodi crittografici?

Manipolazione dei parametri:

- In che modo un aggressore potrebbe manipolare i parametri per influenzare la logica di protezione implementata sul server?
- Come potrebbe un utente malintenzionato manipolare i dati sensibili presenti nei parametri?

Gestione delle eccezioni: