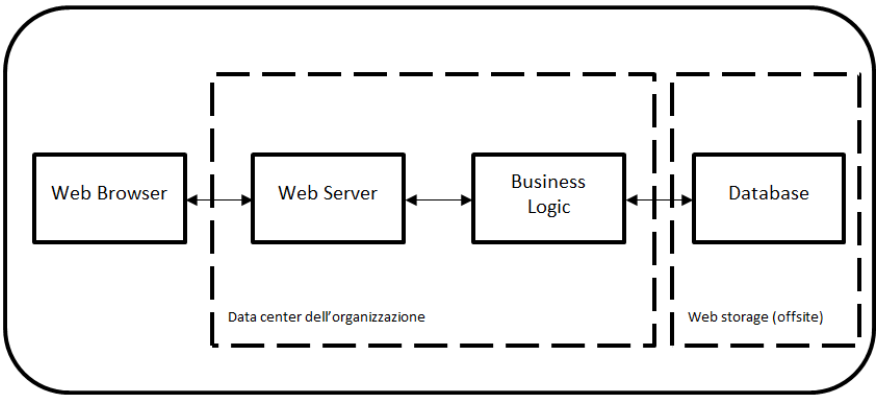

Figura 6 - Aggiunta dei "Trust boundaries" al diagramma

Quando il diagramma diventa più grande e più complesso, può essere molto utile numerare ogni processo, flusso dati e archivio dati presenti nel diagramma, come mostra la figura che segue (Ciascun "trust boundary" dovrebbe avere un identificativo univoco in rappresentanza del suo contenuto):

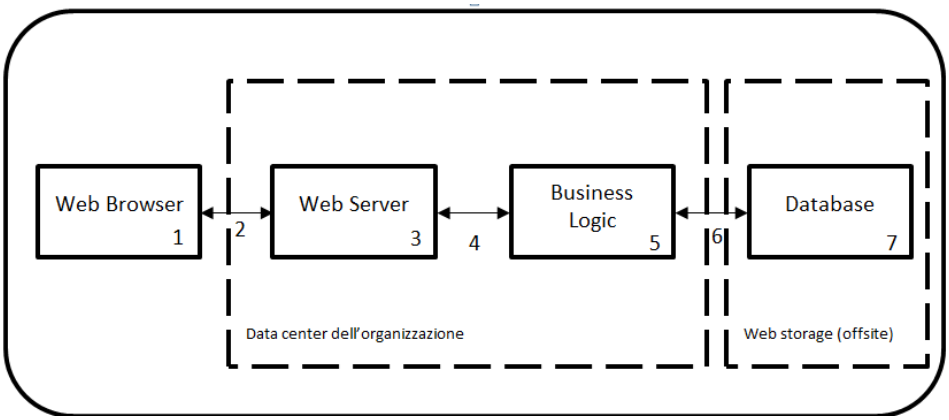

Figura 7 - Numerazione degli elementi del diagramma

Si deve pensare al diagramma del modello come parte integrante del processo di sviluppo, quindi deve essere messo sotto il controllo di versionamento così come tutto il resto del materiale relativo al progetto. Da questo modello, si procede con l'individuazione delle minacce sulla base delle metodologie e delle tecniche descritte nel paragrafo successivo.

5.5.4 Tecniche di modellazione e individuazione delle minacce

L'obiettivo che tutte le metodologie di modellazione delle minacce condividono è lo sviluppo di un processo di passi iterativi che un team può facilmente seguire durante la valutazione di un sistema software.

5.5.4.1 Microsoft SDL – STRIDE

La STRIDE è un processo metodologico che aiuta a individuare le minacce di sicurezza in un sistema complesso. L'elemento mnemonico STRIDE è acronimo di Spoofing, Tampering, Repudiation, Information disclosure, Denial of service ed Elevation of privilege.
Si riporta a seguire la descrizione delle classi di minaccia rappresentate dalla STRIDE:

- **Spoofing** (falsificazione di identità): è la pretesa di essere qualcos'altro o qualcun altro che non si è. Classifica l'insieme delle minacce che consentono a un attaccante di interagire con il sistema

utilizzando un'altra identità. Per esempio, un server di phishing che pretende di essere il server della nostra banca.

- **Tampering** (manomissione dei dati): è l'alterazione di qualcosa che si presuppone non sia oggetto di modifica. Ciò può includere pacchetti di rete (sia fissa che mobile), dati persistiti su supporto di massa o in memoria nonché il codice applicativo. Classifica l'insieme delle minacce che permettono di modificare in modo fraudolento, dati e codice delle applicazioni. Per esempio, un attaccante sfruttando un bug software riesce a cambiare il codice di un'applicazione per aprire una back-door nel sistema.
- **Repudiation** (ripudio di una azione): significa dichiarare di non aver fatto qualcosa (indipendentemente dal fatto che sia stato fatto o no). Classifica l'insieme delle minacce che permettono ad un attaccante di negare di aver compiuto un'azione sul sistema. Per esempio, un utente compie un'azione illegale sul sistema e il sistema non è in grado di rilevare l'azione o di identificare l'utente.
- **Information Disclosure** (divulgazione di informazioni): riguarda l'esposizione delle informazioni a persone non autorizzate alla loro visione. Classifica l'insieme delle minacce che causano l'esposizione di informazioni ad utenti/individui a cui non è consentito l'accesso in lettura. Per esempio, un utente legge un file per il quale non ha ricevuto i diritti di lettura oppure un attaccante legge i dati in transito sulla rete.
- **Denial of Service** (diniego di servizio): sono attacchi designati all'interruzione del servizio erogato da sistemi. Questi includono come effetto il crashing, il rallentamento che porta alla non usabilità del sistema e il riempimento degli storage. Classifica l'insieme delle minacce che permettono di negare o degradare la fornitura di un servizio. Per esempio, un attaccante invia numerosi pacchetti al fine di ostruire la banda di rete di un server il quale non potrà a sua volta essere contattato e/o fornire i suoi servizi agli utenti legittimi.
- **Elevation of Privilege** (elevazione dei privilegi): avviene quando un programma o un utente è tecnicamente abilitato a fare cose che si presuppone non debba fare. Classifica l'insieme delle minacce che permettono ad un utente di ottenere privilegi non previsti per il suo ruolo. Per esempio, un utente anonimo sfrutta un bug software per ottenere i privilegi di amministratore.

In riferimento al diagramma (Figura 7):

- Come si fa a sapere se il browser web verrà utilizzato solo dalle persone che ci si aspetta?
- Cosa accade se qualcuno altera i dati presenti nel database?
- È corretto far transitare i dati da una componente all'atra del sistema senza che questi vengano cifrati?

Questi sono esattamente e rispettivamente esempi di spoofing, tampering e information disclosure che possono essere facilmente individuati con l'ausilio della STRIDE. Con una scarsa conoscenza della sicurezza, ma con l'impiego delle giuste tecniche, è possibile trovare le minacce più importanti in modo veloce e con maggiore affidabilità. Se si sta impiegando un processo di modellizzazione delle minacce, la documentazione prodotta da tale processo, può incrementare il livello di fiducia nel realizzare un software più sicuro.
Per ciascuna minaccia vengono evidenziati gli elementi del diagramma su cui impattano (normalmente si ha maggiore impatto sul software, sui flussi dati o gli storage piuttosto che sui trust boundary).
La lista che segue fornisce alcuni esempi di minacce per ciascuna categoria (l'elenco non vuole essere esaustivo):

SPOOFING	Qualcuno potrebbe fingere di essere un altro utente del nostro sistema, quindi serve un modo per autenticare tutti gli utenti.

	Qualcuno potrebbe fingere di essere il nostro sito web, quindi è necessario assicurarsi di avere un certificato SSL e di utilizzare un singolo dominio per tutte le nostre pagine (per aiutare quel sottoinsieme di utenti che leggono gli URL per vedere se si trovano nel posto giusto).
	Qualcuno potrebbe mettere un collegamento nascosto in una delle nostre pagine, ad esempio logout.html o placeorder.aspx. Dobbiamo controllare il campo HTTP "Referer" prima di intraprendere qualsiasi azione. Non è una soluzione definitiva per contrastare il CSRF (Cross Site Request Forgery), ma è un inizio.

TAMPERING	Qualcuno potrebbe manomettere i dati del back-end.
	Qualcuno potrebbe manomettere i dati in transito tra il data center e il consumer.
	Chi sviluppa potrebbe rilasciare il codice del front-end dell'applicazione senza verificarlo, pensando che sia in fase di caricamento nell'area di staging. Ciò consentirebbe ad uno sviluppatore malintenzionato di aggiungere codice malevolo.

REPUDIATION	Le azioni precedenti potrebbero richiedere una attenta analisi per comprendere ciò che è accaduto. E' necessario porsi delle domande quali: Esistono i log di sistema? Nel log di sistema vengono registrate le giuste informazioni? Il log di sistema è protetto da tampering?

INFORMATION DISCLOSURE	Cosa accade se qualcuno legge i dati presenti nel database? E' possibile che qualcuno possa connettersi al database per leggere o scrivere informazioni?

DENIAL OF SERVICE	Cosa accade se migliaia di utenti si connettono contemporaneamente alla nostra applicazione? E se il sistema va giù?

ELEVATION OF PRIVILEGE	Magari il front-end è l'unico punto di accesso al nostro sito da parte degli utenti, ma cosa lo impone? Cosa previene l'accesso diretto da parte degli utenti alla logica di business in esecuzione sul server o al caricamento di un nuovo codice? Se è presente un firewall, questo è stato correttamente configurato? Chi controlla l'accesso al database, o cosa accade se un impiegato commette un errore o premeditatamente modifica i file a livello di configurazione?

Microsoft ha inizialmente previsto l'applicabilità della STRIDE solo per i punti di ingresso/uscita del sistema in analisi. Tale approccio è stato poi affinato applicando la STRIDE a tutti gli elementi DFD del modello.
Segue una tabella che indica l'esposizione in termini di vulnerabilità secondo la STRIDE rispetto agli elementi DFD utilizzati nel processo di modellazione:

Elemento DFD	S	T	R	I	D	E
Entità Esterna	**X**		**X**			
Flusso Dati		**X**		**X**	**X**	
Archivio Dati		**X**	*	**X**	**X**	
Processo	**X**	**X**	**X**	**X**	**X**	**X**