# 7 UN ESEMPIO APPLICATIVO: CASO D'USO "EASY WEB SITE"

Lo use case si riferisce a un classico sito web disponibile su Internet che implementa un servizio per i propri clienti, i quali vi accedono attraverso un browser.
A titolo di esempio, si suppone che:

- il sito web acceda a una base dati di tipo SQL sia in lettura sia in scrittura;
- il sito web esponga funzionalità sia per i clienti del servizio sia per gli amministratori del servizio;
- l'utenza non autenticata (ad esempio, gli anonymous users) non possa accedere al sistema.

Sulla base delle assunzioni fatte, andiamo a rappresentare il sistema in oggetto attraverso un diagramma, facendo uso del simbolismo DFD, tipicamente utilizzato nella modellazione delle minacce (vedi paragrafo 5.5.3.1). Il diagramma che segue, mostra una scomposizione del sistema in oggetto ponendo in evidenza quelle che sono le sue componenti principali (Browser Client, Web Server e SQL Database), i confini di fiducia (Generic Trust Border Boundary e Internet Boundary) nonché i flussi dati di interscambio tra le singole componenti del sistema ([BC2WS] HTTPS Req (Credentials&Data) - Browser Client to Web Server, [WS2BC] HTTPS Req (Data) - Web Server to Browser Client, [WS2SQLDB] (Credentials&Data) Web Server to SQL Database e [WS2SQLDB] (Data) SQL Database to Web Server) dette anche interazioni.

## 7.1 Diagramma: Use Case

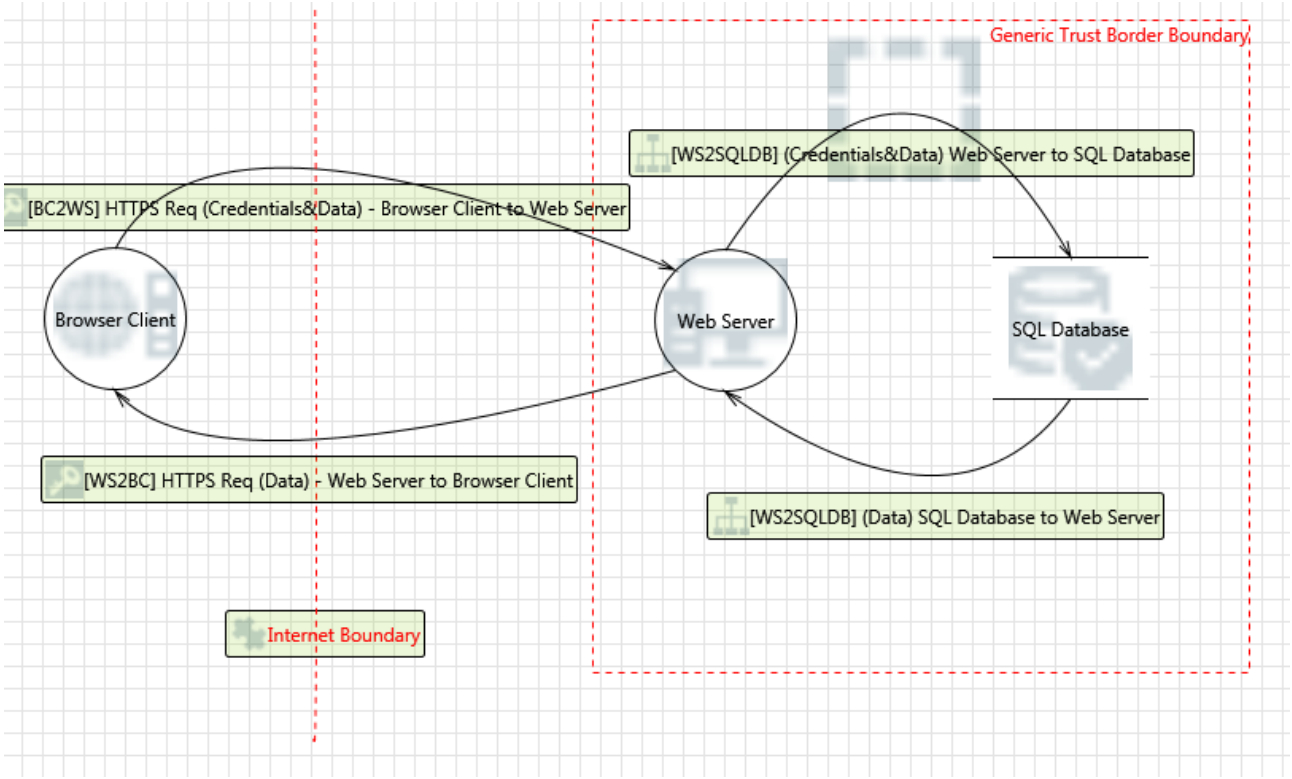


*Figura 9 - Diagramma dello use case*

A seguire, per ciascuna interazione/flusso dati, vengono individuate le possibili minacce sulla base dell'analisi STRIDE. Per ciascuna minaccia viene fornita la categoria STRIDE/Compliance di pertinenza a cui la minaccia appartiene, una breve descrizione e alcune contromisure da attuare nel processo di mitigazione. Viene inoltre indicato, attraverso l'analisi DREAD, un indice di priorità (ALTO, MEDIO e BASSO) da considerare nella risoluzione della minaccia stessa (DREAD Score).

## 7.2 Interazione: da Browser Client a Web Server

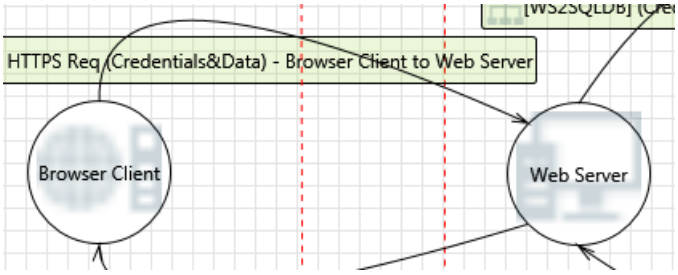


*Figura 10 - Interazione tra Browser Client e Web Server*

### 7.2.1 Assunzioni

Si assume che il Browser Client si autentica nei confronti del Web Server utilizzando una username e una password, ed esegue una post http per leggere e modificare i dati.
Si suppone inoltre, come già detto, che, l'utenza non autenticata (ovvero gli anonymous users) non possa accedere al sistema.
Il protocollo utilizzato è HTTPS, il quale garantisce:

- Autenticazione della Destinazione (Web Server);
- Confidenzialità;
- Integrità.

A seguire vengono riportate le minacce individuabili nell'interazione in oggetto.

### 7.2.2 Accesso a internet non valido

**Categoria:** Compliance

**Descrizione:** L'applicazione Web (qui "Server Web") non dovrebbe essere collegata direttamente a Internet.

**Contromisure:** Interconnettere il "Browser Client" con il "Server Web" tramite un gateway di protezione (firewall).

**Valutazione della priorità della minaccia (Ranking)**

| DREAD | Descrizione | Score |
|---|---|---|
| Damage Potential | Se il Web Server è esposto direttamente sulla rete Internet, un attaccante può facilmente compromettere il sistema. | 3 |
| Reproducibility | L'attacco può essere condotto in qualunque momento. | 3 |
| Exploitability | L'attacco non è banale: occorre una figura senior. | 2 |
| Affected Users | 100%. | 3 |
| Discoverability | Occorre identificare un exploit (es. per mancanza di patching adeguato del Sistema Operativo) cui la macchina su cui il Web Server è installato risulta vulnerabile. | 1 |

**DREAD Score: 12/15 (ALTO)**

### 7.2.3 Mancanza di convalida dell'input da parte del "Web Server"

**Categoria:** Tampering

**Descrizione:** Il 'Web Server' non verifica che i dati di input siano nel formato previsto. Questa è la causa principale di un numero molto elevato di problemi sfruttabili. Considerate tutti i percorsi e il modo in cui si gestiscono i dati di input. Verificare che tutti gli input siano verificati e, se in formato non previsto, scartati o bonificati.

**Contromisure:**

- La validazione dell'input deve essere eseguita prima che l'input entri nel 'Web Server' o venga elaborato da quest'ultimo.
- La convalida dei dati ricevuti deve comprendere almeno:
  - La verifica del tipo di dato;
  - Il controllo dell'intervallo di ammissibilità dei valori dei dati per garantire che i valori forniti siano entro i limiti prestabiliti (minimo e massimo / dimensione);
  - Il controllo sulla base delle regole di business previste.
- Definire il set consentito di caratteri da accettare. La convalida basata su White list è da preferire a quella basata su Black list. La White list prevede la definizione esatta di ciò che è consentito, e per definizione, tutto il resto non è ammesso. L'utilizzo di espressioni regolari può facilitare l'implementazione di schemi di validazione basati su White list. La Black List cerca di rilevare caratteri e modelli di attacco ed è un approccio sconsigliato, in quanto è possibile per un aggressore eludere tali filtri.
- Mettere in campo un meccanismo di controllo degli accessi efficace capace di garantire l'accettazione dei dati solo da parte di utenti autorizzati.

**Valutazione della priorità della minaccia (Ranking)**

| DREAD | Descrizione | Score |
|---|---|---|
| Damage Potential | La mancanza di validazione dell'input da parte del Web Server lo espone a un'ampia varietà di attacchi che possono anche arrivare a compromettere la macchina su cui il Web Server è installato come nel caso di "Command Injection".<br>**NOTA BENE**<br>Si tratta di una minaccia molto generica che vuole focalizzare l'attenzione sul principio "all input is evil". Nel prosieguo vengono esaminate minacce più specifiche legate alla mancanza di validazione dell'input (cross-site scripting, sql injection, ecc.). | 3 |
| Reproducibility | L'attacco può essere condotto in qualunque momento | 3 |
| Exploitability | L'attacco non è banale: occorre una figura senior. | 2 |
| Affected Users | 100% | 3 |
| Discoverability | Occorre identificare un exploit, attraverso la manomissione dei dati di input, cui il Web Server risulta vulnerabile. | 1 |

**DREAD Score: 12/15 (ALTO)**

### 7.2.4 Cross Site Scripting

**Categoria:** Tampering

**Descrizione:** Il 'Web Server' potrebbe essere soggetto ad un attacco di Cross-Site Scripting in quanto non prevede la bonifica dell'input non affidabile (untrusted) che potrebbe contenere script malevoli

**Contromisure:**

- Eseguire l'escape del testo HTML prima di inserire i dati non attendibili nel contenuto degli elementi HTML.
- Eseguire l'escape del valore di un attributo prima di inserire dati non attendibili in attributi HTML.

- Eseguire l'escape del testo JavaScript prima di inserire dati non attendibili nel codice JavaScript.
- Eseguire l'escape HTML di valori JSON prima di inserire i dati nel contenuto degli elementi HTML e leggere i dati con "JSON.parse".
- Eseguire l'escape CSS e attuare rigorose validazioni prima di inserire i dati non attendibili nei valori di proprietà di stile HTML.
- Eseguire l'escape dell'URL prima di inserire dati non attendibili nei valori dei parametri dell'URL.
- Bonificare i Markup HTML con una libreria progettata a tale scopo.
- Utilizzare il flag HTTPOnly per i cookie.
- Implementare la politica Content Security Policy.
- Utilizzare un sistema Auto-Escaping Template System.
- Utilizzare l'X-XSS-Protection Response Header.

**Valutazione della priorità della minaccia (Ranking)**

| DREAD | Descrizione | Score |
|---|---|---|
| Damage Potential | Lo script malevolo può accedere a qualsiasi cookie, token di sessione o altre informazioni sensibili conservate dal browser (qui Browser Client) e utilizzati esclusivamente nel dialogo con il sito d'origine (qui Web Server). Questi script possono anche riscrivere il contenuto della pagina HTML. In definitiva il Tampering dell'url produce Information Disclosure, tra cui la compromissione del token di sessione che abilita il "Session hijacking" (che è una forma di furto di identità – spoofed identity). Nel peggiore dei casi, l'attaccante potrebbe impersonare l'amministratore del Web Server. | 2 |
| Reproducibility | L'attacco funziona sempre. Tuttavia il token di sessione (che è il dato la cui compromissione è particolarmente grave: spoofed identity) è utilizzabile finché la sessione non scade. | 2 |
| Exploitability | L'attacco non è banale: occorre una figura senior. | 2 |
| Affected Users | 100% (nel caso in cui l'attaccante arrivasse a impersonare l'amministratore.) | 3 |
| Discoverability | Occorre identificare un url in cui un input utente ritorna in output senza aver subito alcuna bonifica o che può modificare il "DOM" enviroment. | 1 |

**DREAD Score: 10/15 (MEDIO)**

### 7.2.5 Ripudio di dati da parte del 'Browser Client'

**Categoria:** Repudiation

**Descrizione:** Il 'Client Browser' sostiene di non aver inviato i dati al 'Web Server'.

**Contromisure:**

- È consigliabile che l'applicazione ricevente (qui 'Web Server') utilizzi file di log o di audit per registrare l'origine, l'ora e il riepilogo dei dati ricevuti, affinché il mittente di informazioni non possa negare l'invio delle stesse.
- Si raccomanda inoltre che il destinatario autentichi il mittente per assicurare che la comunicazione avvenga con il mittente corretto.

- Implementare le protezioni contro la manomissione dei dati di log/audit poiché, dati di log/audit manomessi possono produrre potenziali repudiation.
- Considerare di far sì che l'applicazione ricevente richieda al mittente di firmare i dati trasmessi per garantire che il mittente di informazioni non possa negare l'invio delle stesse.

**Valutazione della priorità della minaccia (Ranking)**

| DREAD | Descrizione | Score |
|---|---|---|
| Damage Potential | A fronte della elaborazione dati o dell'esecuzione di transazioni disconosciute dalla sorgente, non si ha modo di attribuire il malfunzionamento alla parte che ne è responsabile. | 1 |
| Reproducibility | L'attacco può essere condotto in qualunque momento. | 3 |
| Exploitability | Per la natura del servizio, l'attacco richiede un'utenza autenticata. | 2 |
| Affected Users | 100% (qualunque utente potrebbe tentare la repudiation). | 3 |
| Discoverability | Il rilevamento della minaccia è contestualizzato nell'ambito di un'utenza autenticata. | 2 |

**DREAD Score: 11/15 (MEDIO)**

### 7.2.6 Crash o fermo del processo 'Web Server'

**Categoria:** Denial Of Service

**Descrizione:** Il 'Web Server' va in crash, si ferma o risponde lentamente, in ogni caso violando una metrica di disponibilità.

**Contromisure:**

- Convalidare tutti i dati di input per assicurare che i valori non possano causare il crash del 'Web Server'.
- Gestire tutti i casi di errore (sia le exceptions del linguaggio di programmazione sia i casi di errore nelle condizioni logiche) in modo graceful, ossia in modo che non provochi crash o servizi degradati.
- I log devono indicare se è in atto o meno una corretta validazione degli input (per evitare crash) e come vengono trattati i casi eccezionali.
- Prevedere il ripristino del sistema: Si consideri l'utilizzo di un meccanismo di recupero (ad esempio il watchdog) per riavviare il 'Web Server' in caso di crash.
- Utilizzare le tecniche di throttling e rate-limiting per evitare che il 'Web Server'collassi.

**Valutazione della priorità della minaccia (Ranking)**

- Si considera lo scenario del DDOS, oggi disponibile "As-a-Service" sul Dark Web.

| DREAD | Descrizione | Score |
|---|---|---|
| Damage Potential | L'attaccante può impedire agli utenti del sistema di interagire con esso (del tutto o in modo degradato). | 2 |
| Reproducibility | L'attacco funziona solo in certe finestre temporali: un attacco DDOS ha per sua natura una durata limitata nel tempo. | 1 |
| Exploitability | L'attacco richiede un figura senior capace di organizzare un DDOS. | 1 |
| Affected Users | 100% (la piattaforma è resa indisponibile o comunque ne viene degradato il funzionamento). | 3 |

| | | |
|---|---|---|
| Discoverability | L'attaccante dovrà impegnare parecchie risorse per sfruttare la vulnerabilità (deve probabilmente sapersi muoversi sul Dark Web e pagare il "servizio"). | 1 |

**DREAD Score (DDOS): 8/15 (MEDIO)**

**Valutazione della priorità della minaccia (Ranking)**

- Si considera lo scenario del crash, nell'ipotesi richieda lo sfruttamento di una vulnerabilità 0-day del 'Web Server'.

| DREAD | Descrizione | Score |
|---|---|---|
| Damage Potential | L'attaccante può impedire agli utenti del sistema di interagire con esso. | 2 |
| Reproducibility | L'attacco funziona fino all'applicazione della patch (che richiede la scoperta dello 0-day, la produzione della patch, il testing della patch, l'installazione della patch). | 2 |
| Exploitability | L'attacco richiede una figura senior capace di trovare una vulnerabilità tale da provocare il crash di Web Server. | 1 |
| Affected Users | 100% (la piattaforma è resa indisponibile o comunque ne viene degradato il funzionamento). | 3 |
| Discoverability | L'attaccante dovrà impegnare parecchie risorse per scoprire la vulnerabilità sfruttabile. Lo sforzo potrebbe non valere il risultato. | 1 |

**DREAD Score (Crash): 9/15 (MEDIO)**
**DREAD Score complessivo (il peggiore tra i due): 9/15 (MEDIO)**

#### 7.2.7 Interruzione del flusso dati HTTPS (o inaccessibilità da parte del 'Web Server')

**Categoria:** Denial Of Service

**Descrizione:** Un agente esterno interrompe i dati che fluiscono attraverso '[BC2WS] HTTPS Req (Credentials&Data) - Browser Client to Web Server' in entrambe le direzioni.

**Contromisure:**

- Se il "Client Browser" non è in grado di proseguire l'elaborazione, dovrebbe fornire risposte appropriate agli utenti in attesa.
- Rilevamento: Si consideri l'utilizzo di un meccanismo di allarme che, rilevando lo spostamento del comportamento del Server Web da metriche predefinite, consenta di segnalare la sospetta condizione di DOS (per attivare le risposte appropriate).
- 3) Ridondanza: considerare la possibilità di configurare un secondo "Web Server" per essere utilizzato come un backup di quello sotto attacco.

**Valutazione della priorità della minaccia (Ranking)**

| DREAD | Descrizione | Score |
|---|---|---|
| Damage Potential | L'attaccante può impedire agli utenti del sistema di interagire con esso. | 2 |
| Reproducibility | Non sembra verosimile che l'attaccante "a comando" / "a piacimento" possa interrompere il flusso dati in qualunque momento. Sembra ragionevole assumere che l'attacco funzioni solo in certe condizioni che si raggiungono raramente. | 1 |

| | | |
|---|---|---|
| Exploitability | L'attacco, sia esso condotto sul piano dell'interruzione fisica della connessione o sul piano dell'interruzione logica del flusso dei dati, è complesso. | 1 |
| Affected Users | 100% (la piattaforma è resa indisponibile). | 3 |
| Discoverability | L'attaccante dovrà impegnare parecchie risorse per organizzare l'attacco. | 1 |

**DREAD Score: 8/15 (MEDIO)**

### 7.2.8 Elevazione di privilegi attraverso l'esecuzione remota di codice da parte del 'Web Server'

**Categoria:** Elevation Of Privilege

**Descrizione:** 'Client Browser' potrebbe essere in grado di eseguire codice in remoto sul sistema ' Web Server'.

**Contromisure:**

- Il processo non deve contenere percorsi che mandano in esecuzione dati presi dal flusso di input (es. il nome di un eseguibile).
- Se un processo manda in esecuzione dati presi dal flusso di input, questi devono essere convalidati in modo da escludere che venga eseguito codice arbitrario.

**Valutazione della priorità della minaccia (Ranking)**

| DREAD | Descrizione | Score |
|---|---|---|
| Damage Potential | L'attaccante potrebbe prendere il controllo dell'intero sistema, attraverso tecniche di "lateral moving" (il "lateral moving" di solito comporta attività legate alla ricognizione <<information gathering>>, furto di credenziali e spostamenti su altri computer). | 3 |
| Reproducibility | L'attacco può essere condotto in qualunque momento. | 3 |
| Exploitability | Per la natura del servizio, l'attacco richiede un'utenza autenticata. Si richiedono anche skill elevati. | 1 |
| Affected Users | 100% (se l'esito finale fosse effettivamente il controllo del sistema). | 3 |
| Discoverability | L'attaccante dovrà impegnare parecchie risorse per scoprire la vulnerabilità sfruttabile (l'attacco di norma sfrutta una catena di debolezze). | 1 |

**DREAD Score: 11/15 (MEDIO)**

### 7.2.9 Elevazione dei privilegi attraverso il cambiamento del flusso di esecuzione nel codice del 'Web Server'

**Categoria:** Elevation Of Privilege

**Descrizione:** Un attaccante può passare dati al 'Web Server' in modo da cambiare a suo vantaggio il flusso di esecuzione del programma all'interno del 'Web Server' stesso.

**Contromisure:**

- Convalidare in modo appropriato gli input e gestire le eccezioni per evitare percorsi di esecuzione imprevisti.
- Impiegare meccanismi di protezione contro il buffer overflow e altri problemi di gestione della memoria.
- Applicare principio del minimo privilegio.

- Implementare le protezioni contro la manomissione dei percorsi di autenticazione e di autorizzazione (ad esempio, se l'autenticazione e l'autorizzazione dipendono dai dati del database, i percorsi di codice che interagiscono con il database devono essere protetti per garantire l'integrità di tali dati).
- Utilizzare implementazioni dell'Address Space Layout Randomization (ASLR) per rendere più difficile l'esecuzione di istruzioni privilegiate agli indirizzi noti in memoria tramite buffer overruns.
- Utilizzare compilatori che bloccano il buffer overruns.

**Valutazione della priorità della minaccia (Ranking)**

| DREAD | Descrizione | Score |
|---|---|---|
| Damage Potential | L'attaccante potrebbe prendere il controllo del Web Server, se riuscisse a elevare i propri privilegi fino a livello appunto di amministratore del Web Server. | 2 |
| Reproducibility | L'attacco può essere condotto in qualunque momento. | 3 |
| Exploitability | Per la natura del servizio, l'attacco richiede un'utenza autenticata. | 2 |
| Affected Users | 100% (se l'esito finale fosse effettivamente il controllo del sistema). | 3 |
| Discoverability | L'attaccante dovrà impegnare parecchie risorse per scoprire la vulnerabilità sfruttabile. | 1 |

**DREAD Score: 11/15 (MEDIO)**

### 7.2.10 Cross Site Request Forgery

**Categoria:** Elevation Of Privilege

**Descrizione:** Il Cross Site Request Forgery (CSRF o XSRF) è un tipo di attacco in cui un attaccante fa in modo che un utente vittima (qui un Autheticated User del Web Server) invii involontariamente una richiesta HTTPS dal suo browser (qui Client Browser) al sistema web (qui Web Server) dove è attualmente autenticato.
L'attaccante deve: a) trovare un difetto (flaw) lato server (qui Web Server) tale per cui il sito web processa una richiesta di cambio stato a fronte della sola presenza di una sessione valida (che attesta una precedente autenticazione); b) indurre un utente ignaro (qui un Autheticated User del Web Server) ad esercitare un url che sfrutta il difetto di cui sopra mentre quell'utente ha una sessione aperta sul server (qui Web Server).
Il sistema, vulnerabile al CSRF, riceve dal browser dell'utente la richiesta contraffatta (dietro cui, cioè, si cela un'azione studiata dall'attaccante) con un cookie di sessione valido (dal momento che la vittima è stata precedentemente autenticata e la sessione è ancora attiva) e la elabora.

**Contromisure:** Fare in modo che tutte le richieste di cambiamento di stato oltre ad essere autenticate includano un ulteriore elemento di payload segreto (canary o CSRF token) conosciuto solo dal sito legittimo e dal browser (parti che comunicano tra loro in modo protetto tramite HTTPS).

**Valutazione della priorità della minaccia (Ranking)**

| DREAD | Descrizione | Score |
|---|---|---|

| | | |
|---|---|---|
| Damage Potential | L'attaccante può eseguire richieste di cambio stato al sistema che sono prerogativa di Autheticated User o, peggio, dell'Administrator (es. la modifica di configurazioni o il furto/modifica di dati privati). | 2 |
| Reproducibility | L'attacco funziona finché la sessione della vittima non scade. | 1 |
| Exploitability | L'attacco non è banale: occorre una figura senior. | 2 |
| Affected Users | In genere l'attacco è condotto con tecniche di social engineering: gli utenti coinvolti sono una quota parte del totale. | 2 |
| Discoverability | Occorre identificare un url che presenti la vulnerabilità XSRF. | 1 |

**DREAD Score: 8/15 (MEDIO)**

## 7.3 Interazione: da Web Server a Browser Client

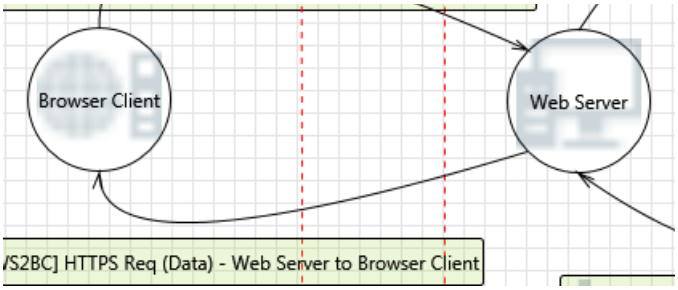


*Figura 11 - Interazione tra Web Server e Browser Client*

### 7.3.1 Assunzioni

Si assume che il Browser Client si autentica nei confronti del Web Server utilizzando una username e una password, ed esegue una post http per leggere e modificare i dati.
Si suppone inoltre, come già detto, che, l'utenza non autenticata (ovvero gli anonymous users) non possa accedere al sistema.
Il protocollo utilizzato è HTTPS, il quale garantisce:

- Autenticazione della Destinazione (Web Server);
- Confidenzialità;
- Integrità.

### 7.3.2 Analisi delle minacce e mitigazioni

Valgono le raccomandazioni già proposte in "Interazione: da Browser Client a Web Server".

## 7.4 Interazione: da Web Server a SQL Database

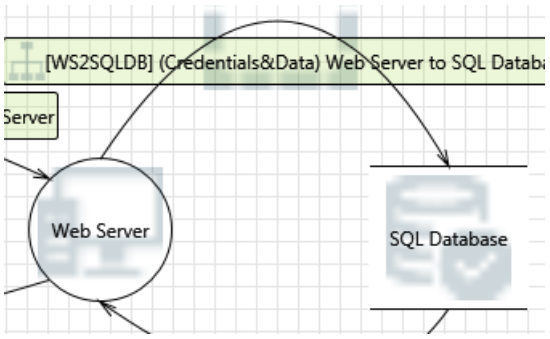


*Figura 12 - Interazione tra Web Server e SQL Database*

### 7.4.1 Assunzioni

Si assume che il Web Server si autentica nei confronti del SQL Server utilizzando una username e una password, e può inserire, leggere, modificare, cancellare dati.
Si suppone che l'interazione avvenga all'interno di un Trusted Boundary.
A seguire vengono riportate le minacce individuabili nell'interazione in oggetto.

### 7.4.2 Vulnerabilità di SQL Injection nel 'SQL Database'

**Categoria:** Tampering

**Descrizione:** La SQL Injection è una tecnica di attacco di tipo "code injection", usata per attaccare un database, nella quale viene inserito del codice SQL malevolo all'interno di parametri di input in modo che vada in esecuzione sul database sotto attacco (es. per inviare all'attaccante /modificare/distruggere il contenuto del database e, in certi casi, "saltare dal DB al Sistema Operativo" e prendere il controllo della macchina). A patto che la query risultante da questo tipo di attacco sia sintatticamente corretta, il database la eseguirà.

**Contromisure:**

- Usare i Prepared Statements con query parametrizzate.
- Usare le Stored Procedures.
- Eseguire la validazione di tipo White List di ogni input esterno usato per costruire statements SQL.
- 4) Eseguire l'escape di ogni input utente.

**Valutazione della priorità della minaccia (Ranking)**

| DREAD | Descrizione | Score |
|---|---|---|
| Damage Potential | La possibilità di eseguire codice malevolo sul database dell'applicazione la espone potenzialmente alla totale compromissione. | 3 |
| Reproducibility | L'attacco può essere condotto in qualunque momento. | 3 |
| Exploitability | Per la natura del servizio, l'attacco richiede un'utenza autenticata. | 2 |
| Affected Users | 100%. | 3 |
| Discoverability | Occorre identificare un exploit, attraverso la manomissione dei dati di input, cui il Web Server risulta vulnerabile. | 1 |

**DREAD Score: 12/15 (MEDIO)**

### 7.4.3 Possibile compromissione del 'SQL Database'

**Categoria:** Tampering

**Descrizione:** Assicurare l'integrità dei dati all'interno dell'archivio 'SQL Database'.

**Contromisure:**

- Autenticare tutti gli utenti.
- Mettere in pratica un efficace meccanismo di controllo degli accessi che garantisca che i dati possono essere scritti o modificati solo da utenti autorizzati.
- Rispettare il principio del privilegio minimo.
- Attuare controlli che seguano e registrino correttamente le azioni degli utenti.

**Valutazione della priorità della minaccia (Ranking)**

| DREAD | Descrizione | Score |
|---|---|---|

| | | |
|---|---|---|
| Damage Potential | La possibilità, senza averne diritto, di modificare i dati all'interno del database dell'applicazione la espone potenzialmente alla totale compromissione. | 3 |
| Reproducibility | L'attacco può essere condotto in qualunque momento. | 3 |
| Exploitability | Per la natura del servizio, l'attacco richiede un'utenza autenticata. | 2 |
| Affected Users | 100%. | 3 |
| Discoverability | Occorre identificare un exploit, attraverso la manomissione dei dati di input, cui il Web Server risulta vulnerabile. | 1 |

**DREAD Score: 12/15 (MEDIO)**

#### 7.4.4 Consumo eccessivo di risorse da parte del 'Web Server' o del 'SQL Database'

**Categoria:** Denial Of Service

**Descrizione:** Il "Web Server" o il "SQL Database" adottano passi espliciti per controllare il consumo di risorse? Fare attenzione che le richieste di risorse non producano deadlock e che, nel caso peggiore, vadano in timeout.

**Contromisure:**

- Non bloccare (deadlock) le richieste di risorse.
- Impostare i timeout per le richieste di risorse, se è applicabile.
- Validare i dati di input che si riferiscono al consumo di risorse.
- Limitare la dimensione dei dati elaborati dall'applicazione.
- Eseguire il rilascio delle risorse quando non sono più necessarie.
- Gli audit devono dare indicazioni sul consumo eccessivo di risorse da parte dell'applicazione.

**Valutazione della priorità della minaccia (Ranking)**

| DREAD | Descrizione | Score |
|---|---|---|
| Damage Potential | L'attaccante può degradare le prestazioni del sistema fino a renderlo potenzialmente indisponibile. | 2 |
| Reproducibility | L'attacco funziona sempre. | 3 |
| Exploitability | Per la natura del servizio, l'attacco richiede un'utenza autenticata. | 2 |
| Affected Users | 100% (la piattaforma è resa indisponibile o comunque ne viene degradato il funzionamento). | 3 |
| Discoverability | Il rilevamento della minaccia è contestualizzato nell'ambito di un'utenza autenticata. | 2 |

**DREAD Score (Crash): 12/15 (ALTO)**

## 7.5 Interazione: da SQL Database a Web Server

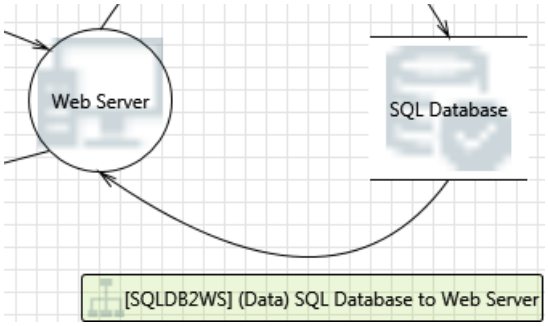


*Figura 13 - Interazione tra SQL Database e Web Server*

### 7.5.1 Assunzioni

Il Web Server si autentica nei confronti del SQL Database utilizzando una username e una password ed inserisce, legge, modifica e cancella dati.
Si suppone che l'interazione avvenga all'interno di un Trusted Boundary.
A seguire vengono riportate le minacce individuabili nell'interazione in oggetto.

### 7.5.2 Persistent Cross Site Scripting

**Categoria:** Tampering

**Descrizione:** Il 'Server Web' potrebbe essere soggetto ad un attacco di cross-site scripting di tipo persistente in quanto non bonifica i dati di input al 'SQL Database' in fase di scrittura (che potrebbero contenere uno script malevolo) e non esegue l'escape dei dati di output dal 'SQL Database' in fase di lettura (ciò che si traduce nel mandare in esecuzione su 'Browser Client' lo script malevolo).

**Contromisure:** Applicare le tecniche di bonifica e di escaping come nel caso di Cross Site Scripting.

**Valutazione della priorità della minaccia (Ranking)**

| DREAD | Descrizione | Score |
|---|---|---|
| Damage Potential | Lo script dannoso può accedere a qualsiasi cookie, token di sessione o altre informazioni sensibili conservate dal browser (qui Browser Client) e utilizzati esclusivamente nel dialogo con il sito d'origine (qui Web Server). Questi script possono anche riscrivere il contenuto della pagina HTML. In definitiva il Tampering dell'url produce Information Disclosure, tra cui la compromissione del token di sessione che abilita il "Session hijacking" (che è una forma di furto di identità – spoofed identity). Nel caso pessimo, l'attaccante potrebbe impersonare l'amministratore del Web Server. | 2 |
| Reproducibility | L'attacco funziona sempre. Tuttavia il token di sessione (che è il dato la cui compromissione è particolarmente grave: spoofed identity) è utilizzabile finché la sessione non scade. | 2 |
| Exploitability | L'attacco non è banale: occorre una figura senior. | 2 |
| Affected Users | 100% (nel caso in cui l'attaccante arrivasse a impersonare l'amministratore). | 3 |

| | | |
|---|---|---|
| Discoverability | Occorre identificare un url che restituisca in output, senza aver subito alcun encoding, un input utente malevolo, precedentemente persistito sul database senza alcuna bonifica. | 1 |

**DREAD Score: 10/15 (MEDIO)**

### 7.5.3 Controllo accesso debole per una risorsa

**Categoria:** Information Disclosure

**Descrizione:** Una inadeguata protezione dei dati a livello di " SQL Database" può consentire a un attaccante di leggere informazioni non destinate alla divulgazione. Esaminare le impostazioni di autorizzazione.

**Contromisure:**

- Autenticare tutti gli utenti.
- Mettere in pratica un efficace meccanismo di controllo degli accessi che garantisca che i dati possono essere letti solo da utenti autorizzati.
- Rispettare il principio del minimo privilegio.
- Attuare controlli che seguano correttamente e registrino le azioni degli utenti
- Crittografare i dati.
- Assicurarsi che le utilità o le tecniche di riservatezza dei data store vengano appropriatamente utilizzate in modo che la riservatezza dei dati sia mantenuta e gestita in base alle esigenze aziendali/regole.

**Valutazione della priorità della minaccia (Ranking)**

| DREAD | Descrizione | Score |
|---|---|---|
| Damage Potential | La possibilità, senza averne diritto, di leggere i dati all'interno del database dell'applicazione espone potenzialmente l'owner del sistema a violazioni di normative di legge (es. Privacy) o a danno reputazionale o a divulgazione di informazioni di business riservate. | 2 |
| Reproducibility | L'attacco può essere condotto in qualunque momento. | 3 |
| Exploitability | Per la natura del servizio, l'attacco richiede un'utenza autenticata. | 2 |
| Affected Users | 100%. | 3 |
| Discoverability | Occorre identificare un exploit, attraverso la manomissione dei dati di input, cui il Web Server risulta vulnerabile. | 1 |

**DREAD Score: 11/15 (MEDIO)**