

UNIVERSITÀ DEGLI STUDI DI NAPOLI
PARTHENOPE

DIPARTIMENTO DI INGEGNERIA



**CORSO DI LAUREA IN INGEGNERIA E SCIENZE INFORMATICHE
PER LA CYBERSECURITY**

PROJECT WORK

**Titolo progetto:
CyberDefender**

DOCENTE

PROF. LUGI COPPOLINO

PROF. LUGI ROMANO

GRUPPO

Filomena Antonietta Assunta Auricchio, 0334000145

Lara Maggiulli, 0334000090

Alessandro Tafuro, 0334000086

ANNO ACCADEMICO 2024/2025

SCHEDA PROGETTO

Tema:

Nome del progetto: CyberDefender

Slogan: “Conosci, Apprendi, Difendi”

Logo del progetto: (LOGO SCUDO)



Composizione del team:

Nome e Cognome	Matricola	Email istituzionale	Email privata	Recapito telefonico
Filomena Antonietta Assunta Auricchio	0334000145	filomenaantoniettaassunta.auricchio001@studenti.uniparthenope.it	filomena.auricchio04@gmail.com	3313825035
Lara Maggiulli	0334000090	lara.maggiulli001@studenti.uniparthenope.it	lara.mag46@gmail.com	3296631244
Alessandro Tafuro	0334000086	alessandro.tafuro001@studenti.uniparthenope.it	tafuro561@gmail.com	3426466250

CyberDefender

1. Cos'è CyberDefender?

CyberDefender è una web application interattiva progettata per informare, educare e supportare gli utenti nel riconoscimento e nella prevenzione degli attacchi informatici più comuni. Essa nasce con lo scopo di:

- Analizzare gli input dell'utente;
- Fornire la conoscenza degli attacchi informatici più diffusi e della sicurezza informatica;
- Fornire strumenti per la risoluzione delle problematiche caricate e consigli educativi per riconoscere e contrastare le minacce;
- Collegare teoria e pratica, guidando l'utente passo dopo passo “*Conosci. Apprendi. Difendi*”.

Descrizione generale del progetto

CyberDefender è una web app pensata per aiutare privati e aziende a riconoscere e prevenire i principali attacchi informatici. Attraverso una serie di strumenti interattivi, la piattaforma fornisce informazioni utili, linee guida pratiche e form guidati per identificare le minacce digitali più comuni e suggerire come difendersi.

Gli utenti possono registrarsi, accedere a contenuti mirati e ricevere supporto personalizzato in base alle proprie esigenze, questo ultima funzionalità in un'ottica futura. La web app include anche funzioni per analizzare messaggi sospetti, verificare numeri di telefono.

L'obiettivo di CyberDefender è rendere la sicurezza informatica più accessibile e comprensibile per tutti, offrendo un aiuto concreto nella gestione dei rischi digitali.

Target degli utenti

La web application è pensata sia per utenti privati che per aziende, offrendo contenuti didattici e funzionalità interattive che li aiutano a comprendere i rischi della cybersicurezza e a difendersi in modo consapevole.

Utente Privato

L'utente privato è tipicamente una persona che:

- Naviga spesso su internet (social media, email, e-commerce, home banking);
- Ha poca o media conoscenza della sicurezza informatica;
- È interessato a imparare a proteggersi da attacchi comuni (phishing, furto credenziali, malware).

Obiettivi per questa tipologia:

- Educazione e sensibilizzazione: rendere consapevole l'utente delle minacce reali nel web quotidiano;
- Prevenzione personale: fornire strumenti semplici per riconoscere un pericolo.

Azienda

Il profilo aziendale è destinato a figure che:

- Lavorano in contesti dove è cruciale la sicurezza di dati e sistemi;
- Devono formare dipendenti o collaboratori;
- Hanno bisogno di strumenti per valutare il rischio interno.

Obiettivi per questa tipologia:

- Formazione del personale
- Identificazione dei punti deboli nel comportamento digitale dei dipendenti

Funzionalità Principali

La piattaforma propone le seguenti funzionalità principali:

1. **Enciclopedia degli attacchi:** Una raccolta organizzata di schede informative che descrivono i principali tipi di attacchi informatici (come phishing, SQL injection, cross-site scripting, ecc.), presentate con un linguaggio chiaro e accessibile a utenti non esperti. Ogni scheda include:
 - a. Definizione dell'attacco;
 - b. Modalità di esecuzione;
 - c. Possibili conseguenze;
 - d. Consigli pratici per la prevenzione.
2. **Compilazione modulo rilevamento attacco:** Sulla base delle informazioni presenti nell'Enciclopedia degli attacchi, l'utente può compilare un modulo guidato per rilevare un possibile attacco informatico in corso. Il modulo propone una serie di domande semplici e mirate, costruite in relazione ai sintomi comuni degli attacchi descritti.
3. **Report PDF con consigli su misura:** Al termine dell'analisi delle risposte, CyberDefender genera un report personalizzato in formato PDF. Questo documento riassume i risultati, individua eventuali vulnerabilità o rischi, e offre una serie di raccomandazioni pratiche per migliorare la propria sicurezza digitale.
4. **Analizzatore di messaggi sospetti:** Uno strumento interattivo che permette agli utenti di inserire messaggi o query sospette (ad esempio email, testi o codici). Il sistema analizza il contenuto alla ricerca di parole chiave o pattern tipici di attacchi informatici, valutando la gravità del rischio e fornendo un feedback immediato. Questo aiuta l'utente a riconoscere tempestivamente potenziali minacce.
5. **Verifica numeri di telefono:** È presente anche una funzionalità per inserire numeri di telefono sospetti e verificarne l'autenticità, distinguendo tra numeri affidabili e potenziali chiamate pubblicitarie o fraudolente.

Funzionalità future specifiche per utenti aziendali:

6. **Accesso a contenuti tecnici più approfonditi:** Per le aziende e i referenti tecnici, CyberDefender mette a disposizione sezioni dedicate con materiale più specialistico, tra cui:
 - a. Linee guida avanzate sulle configurazioni di sicurezza;
 - b. Check-list basate su standard riconosciuti come OWASP;

c. Risorse per la formazione continua del personale in ambito cybersecurity.

Queste funzionalità sono pensate per rispondere alle diverse esigenze degli utenti, garantendo un'esperienza personalizzata e una copertura completa sia degli aspetti teorici che pratici della sicurezza informatica.

Product Vision

CyberDefender nasce dall'esigenza di fornire risposte concrete alle tante domande che gli utenti si pongono in merito alla sicurezza, soprattutto nell'uso quotidiano delle tecnologie digitali. L'obiettivo a breve termine della web app è sensibilizzare gli utenti sui principali rischi informatici e accompagnarli verso un'esperienza digitale più consapevole e sicura. Per farlo, CyberDefender mette a disposizione diversi strumenti: dai form, che aiutano il sistema a identificare potenziali attacchi sulla base delle risposte fornite, fino a funzionalità che permettono di analizzare messaggi sospetti o verificare la natura di numeri di telefono ricevuti tramite chiamata o SMS. Inoltre, la piattaforma permetterà in futuro agli utenti di contattare direttamente il team per ricevere supporto personalizzato. Le risposte possono essere fornite via email oppure attraverso una sezione dedicata all'interno dell'area personale. L'applicazione sarà disponibile gratuitamente. Solo nel caso in cui l'utente desideri usufruire di un confronto più frequente e approfondito con gli esperti, sarà possibile attivare un abbonamento.

Elevator Pitch

CyberDefender è una web app pensata per aiutare privati e aziende a riconoscere e prevenire le minacce informatiche più comuni. Attraverso form per il rilevamento degli attacchi sulla base delle osservazioni riportate dagli utenti, analisi di messaggi sospetti e verifica di numeri di telefono, offre strumenti semplici e concreti per migliorare la sicurezza digitale. L'applicazione è progettata per garantire la massima sicurezza nell'interazione con i dati degli utenti, grazie all'uso di tecnologie moderne che prevengono attacchi informatici e proteggono la privacy. CyberDefender è accessibile, gratuita e supporta l'utente con consigli personalizzati, rendendo la sicurezza online più vicina e comprensibile per tutti.

Obiettivi principali

Gli obiettivi che CyberDefender si propone di raggiungere sono:

- **Educare e istruire gli utenti**, offrendo loro una maggiore consapevolezza sulle tecnologie digitali e sui principali attacchi informatici attualmente più frequenti, per garantire un'esperienza online più sicura.
- **Coinvolgere attivamente gli utenti nella propria sicurezza digitale**, permettendo loro di osservare e analizzare eventuali anomalie per riconoscere possibili attacchi. Sulla base di queste osservazioni, gli utenti possono compilare i form forniti dalla piattaforma, che restituiscono informazioni dettagliate sull'attacco rilevato e offrono linee guida pratiche per proteggersi.

- **Fornire una visione completa degli attacchi subiti**, salvando ogni evento rilevato nell'area personale dell'utente, che può così consultare e analizzare i report relativi.
- **Rassicurare gli utenti riguardo alla natura di messaggi e chiamate ricevuti**, mettendo a disposizione strumenti pratici per proteggerli da eventuali rischi.

Requisiti generali dettagliati

I requisiti generali dettagliati del progetto includono:

- Sviluppo di un applicazione web, con backend realizzato in Python tramite il framework Django, e frontend sviluppato con Bootstrap (HTML, CSS, JavaScript), utilizzando tecnologie moderne e sicure.
- Gestione differenziata degli utenti (privati e aziende) tramite un sistema di controllo degli accessi basato sui ruoli (RBAC), che consente di assegnare permessi specifici di accesso, modifica e visualizzazione per ogni sezione dell'applicazione.
- Adozione di pratiche di sicurezza per prevenire vulnerabilità comuni, tra cui SQL Injection, Cross-Site Request Forgery (CSRF), e gestione sicura dei cookie di sessione, seguendo le linee guida OWASP.
- Implementazione delle funzionalità principali, tra cui form per il riconoscimento di attacchi, analisi semantica dei messaggi sospetti, verifica dei numeri di telefono e generazione di report personalizzati nell'area utente.
- Sistema avanzato di logging e auditing, in grado di registrare tutte le operazioni critiche (es. login, invio richieste, consultazione report) in modo sicuro, tracciabile e immodificabile.
- Esecuzione di test sistematici e approfonditi (unit test, integration test, system test, acceptance test e penetration test) per garantire qualità, sicurezza e affidabilità dell'applicazione.
- Applicazione di strategie di sicurezza basate sul modello STRIDE, per garantire la copertura contro minacce come spoofing, tampering, repudiation, information disclosure, denial of service ed elevation of privilege, in linea con gli standard ISO/IEC 27001, NIST e OWASP ASVS (Application Security Verification Standard).

2. Organizzazione Agile e Sprint

Metodologia Agile adottata

Per consentire uno sviluppo semplice ma efficace del sistema di CyberDefender si è pensato di utilizzare un **approccio AGILE**. Questo perché l'intero team è composto da poche persone che puntano sul coordinamento come caratteristica principale. In più, i requisiti non sono descritti nello specifico e necessitano di essere approfonditi durante le fasi di testing, con l'aiuto delle user stories. Premesso ciò, il team ha deciso di utilizzare una metodologia a **sprint** per assicurarsi un costante flusso di feedback e flessibilità in fase di sviluppo.

Per quanto riguarda la sicurezza, ogni sprint tratta user stories con un focus sulla sicurezza del sistema stesso e alla fine di ogni sprint viene aggiornato il **threat model**. Il lavoro sarà svolto in tre sprint, con una durata di ogni sprint pari ad una settimana.

Ad ogni sprint vengono svolte le seguenti attività:

- Breve meeting giornaliero in cui si descrive l'andamento delle task di ogni membro del team
- Pianificazione dello sprint e suddivisione del lavoro in task da assegnare ai membri
- Showcase delle nuove funzionalità implementate
- Raccolta feedback
- Analisi del lavoro finito con la finalità di migliorare l'intero processo in futuro e il lavoro già svolto nel presente

Panoramica Sprint

Sprint	Durata	User Stories collegate	Obiettivo	Consegnabile
Sprint 1	1 settimana	US09	Definizione e realizzazione del DB. Popolazione del DB con tutti i dati dell'enciclopedia. Implementazione dell'enciclopedia sul sito web	Database completamente funzionante e operativo. Enciclopedia online e consultabile, senza la possibilità di registrare le consultazioni, per mancanza funzionalità login
Sprint 2	1 settimana	US01, US06, US07, US08	Implementazione di pagina registrazione e login. Realizzare analizzatore di messaggi e numeri di telefono.	Funzionalità implementate con successo

Sprint	Durata	User Stories collegate	Obiettivo	Consegnabile
Sprint 3	1 settimana	US03, US04, US05	Implementare la funzionalità di rilevamento attacco e report via documento PDF	Funzionalità implementate con successo

Tabella 1: Panoramica Sprint

Dettaglio Sprint

Sprint 1

Obiettivo: Definire il modello ER e modello logico del database e realizzare l'intero database su di essi. Successivamente, popolare il database con tutti i dati dell'enciclopedia e realizzare una prima interfaccia grafica piacevole ed intuitiva sul sito web. Infine, implementare l'enciclopedia.

Attività realizzate: Il database è stato realizzato e sono presenti alcuni dati, l'interfaccia grafica per visualizzare i dati dell'enciclopedia è stata realizzata e l'enciclopedia è consultabile.

Sprint 2

Obiettivo: Realizzare funzionalità di registrazione e login. Realizzare la funzionalità di analisi dei messaggi e numeri di telefono. Esecuzione dei primi test generali relativi alla sicurezza e test specifici per ogni funzionalità con lo scopo di valutarne le performance.

Attività realizzate: Le funzionalità sono on-line e sono completamente funzionanti. Sono state implementate una versione dell'analizzatore di messaggi e una versione dell'analizzatore di numeri telefonici che hanno passato i primi test generali sul funzionamento e sulla sicurezza.

Sprint 3

Obiettivo: Implementare la funzionalità di rilevamento attacco e report via documento PDF.

Attività realizzate: Le funzionalità sono state implementate e testate ampiamente.

Conclusioni Sprint con ROADMAP

Grazie all'utilizzo della metodologia Agile con sprint, lo sviluppo del sistema è rimasto coerente con i requisiti precedentemente stabiliti e con quelli che si sono evoluti nel tempo, riuscendo anche a testare continuamente la sicurezza e l'efficacia delle funzionalità implementate, anche grazie al continuo ciclo di feedback.

Roadmap Sprint – CyberDefender



3. Analisi dei requisiti

L'analisi dei requisiti rappresenta una fase fondamentale nella progettazione di qualsiasi sistema software, poiché consente di definire con precisione le funzionalità e le caratteristiche che il sistema dovrà offrire. Questa attività ha l'obiettivo di garantire che il prodotto finale soddisfi pienamente le aspettative del committente e risponda in modo efficace alle reali necessità degli utenti.

Nel caso del progetto **CyberDefender**, questa fase assume un'importanza ancora maggiore, poiché la piattaforma è destinata a un ambito particolarmente delicato come la cybersicurezza. È quindi essenziale identificare con accuratezza tutti i requisiti funzionali (le funzionalità che il sistema deve implementare) e i requisiti non funzionali (prestazioni, sicurezza, usabilità, scalabilità, ecc.).

CyberDefender nasce con l'obiettivo di fornire una web application moderna, accessibile e sicura. Questa analisi dei requisiti ha quindi lo scopo di raccogliere e formalizzare un insieme di specifiche che guidino lo sviluppo di una piattaforma:

- **Intuitiva**, per garantire una user experience efficace anche a utenti non esperti;
- **Scalabile**, per supportare l'evoluzione del sistema e la crescita del numero di utenti;
- **Sicura**, secondo i più recenti standard in ambito software e cybersecurity.

L'analisi che segue descrive in dettaglio i requisiti funzionali e non funzionali identificati per la realizzazione del progetto, al fine di garantirne l'efficacia, la sicurezza e l'aderenza ai suoi obiettivi.

Tecniche di Elicitazione dei Requisiti

Per identificare i requisiti di CyberDefender, sono state adottate diverse tecniche di elicazione, con l'obiettivo di comprendere in modo approfondito le esigenze degli utenti finali e degli stakeholder coinvolti nel progetto. Le principali tecniche utilizzate sono le seguenti:

- **Interviste con utenti e stakeholder:** Sono stati condotti colloqui mirati con esperti di cybersecurity e potenziali utenti finali (privati e aziendali). Le interviste hanno permesso di approfondire le esperienze dirette con attacchi informatici, raccogliere esigenze specifiche e individuare le funzionalità più utili in una piattaforma di supporto alla sicurezza digitale.
- **Osservazione diretta:** È stata svolta un'attività di osservazione delle interazioni degli utenti con piattaforme analoghe già presenti sul mercato. Questa fase ha permesso di individuare criticità ricorrenti, lacune funzionali e punti di miglioramento in termini di usabilità e chiarezza dei contenuti.
- **Workshop di brainstorming:** Sono state organizzate sessioni collaborative tra sviluppatori e professionisti della sicurezza informatica, al fine di generare idee innovative e definire le funzionalità essenziali per una piattaforma versatile, in grado di rispondere a una varietà di scenari e minacce digitali.
- **Sondaggi:** Per ottenere un riscontro quantitativo e qualitativo da un pubblico più ampio, sono stati svolti sondaggi. Le domande hanno riguardato abitudini di navigazione, consapevolezza in ambito cybersecurity e aspettative nei confronti di uno strumento educativo e di prevenzione.

Classificazione dei Requisiti

I requisiti di un sistema software possono essere suddivisi in tre categorie principali:

Tipo di Requisito	Descrizione	Esempi nel progetto CyberDefender
Requisiti Funzionali	Definiscono le funzionalità specifiche che il sistema deve offrire agli utenti.	Registrazione utenti, analisi messaggi sospetti, verifica numeri, generazione report PDF, gestione ruoli (RBAC).
Requisiti Non Funzionali	Specificano vincoli e qualità attese nel comportamento del sistema.	Prestazioni, scalabilità, usabilità, sicurezza, integrità dei dati, hashing dati, tempo di risposta.
Requisiti di Dominio	Includono vincoli derivanti dal contesto normativo, tecnico o di settore.	Conformità al GDPR, NIS2, adozione di standard ISO27001, auditing dei log (registro accessi), separazione dei privilegi.

Tabella 2: Tipi di requisiti

CyberDefender incorpora anche funzionalità tipiche di:

- **IDS (Intrusion Detection System)**: attraverso il riconoscimento automatico di pattern sospetti ad esempio nei messaggi/testi inseriti dagli utenti;
- **EPP (Endpoint Protection Platform)**: in quanto fornisce strumenti educativi e pratici per la protezione dell'utente finale.

Grazie a questa integrazione, il progetto si propone non solo come piattaforma informativa, ma anche come strumento attivo di supporto alla sicurezza digitale, in grado di adattarsi a diversi contesti d'uso, sia individuali che aziendali.

I **requisiti funzionali** descrivono le operazioni che il sistema CyberDefender deve essere in grado di eseguire per soddisfare le esigenze degli utenti e degli stakeholder. La seguente tabella riassume i principali requisiti, con l'indicazione della priorità, della fonte e dei criteri di accettazione associati.

ID	Descrizione	Priorità	Fonte	Criterio di Accettazione
RF-01	Il sistema deve consentire il login con username e password	Alta	Utenti	Accesso consentito solo con credenziali valide
RF-02	Il sistema deve permettere di impostare permessi di accesso alle aree private	Alta	Utenti	Solo utenti autorizzati possono accedere, secondo i permessi definiti
RF-03	Il download deve essere consentito solo se l'utente ha i permessi adeguati	Alta	Utenti	Accesso negato a utenti non autorizzati
RF-04	Il sistema deve consentire la visualizzazione del registro di controllo relativo agli eventi di sicurezza, con	Alta	Responsabile	Eventi accessibili da dashboard

ID	Descrizione	Priorità	Fonte	Criterio di Accettazione
	particolare riferimento ai tentativi di accesso non riusciti			
RF-05	Il sistema deve essere compatibile con Windows, Linux e macOS (implementazione futura)	Alta	Stakeholder	Il client si installa correttamente su tutti i sistemi supportati
RF-06	Il sistema deve essere scalabile per ambienti con centinaia o migliaia di dispositivi (implementazione futura)	Alta	Stakeholder	Il sistema mantiene performance accettabili con >100 dispositivi
RF-07	Il sistema deve integrare funzioni di auto-apprendimento contro nuove minacce (implementazione futura)	Alta	Stakeholder	Il sistema aggiorna autonomamente le regole comportamentali
RF-08	Il sistema deve permettere il recupero della password tramite email (implementazione futura)	Alta	Utenti	L'utente riceve una email con istruzioni per il reset della password in caso di smarrimento

Tabella 3: Requisiti funzionali

I **requisiti non funzionali** riguardano gli aspetti che non sono direttamente legati alle funzionalità, ma sono fondamentali per garantire l'efficacia, l'affidabilità e la qualità complessiva del sistema. Questi includono performance, sicurezza, usabilità, scalabilità, e altro.

ID	Descrizione	Priorità	Fonte	Acceptance Criteria
RNF-01	Il sistema deve rispondere entro 3 secondi per le operazioni comuni (es. analisi messaggi, numeri sospetti)	Alta	Stakeholder	Test di analisi completati in ≤ 3 secondi
RNF-02	L'applicazione web deve garantire un tempo di disponibilità (uptime) minimo del 99,5%	Alta	Stakeholder	Monitoraggio mensile dell'uptime $\geq 99,5\%$
RNF-03	L'interfaccia utente deve essere chiara, accessibile e reattiva	Media	Utenti	Feedback positivo $\geq 80\%$ su usabilità e semplicità d'uso
RNF-04	Tutti i dati sensibili, come le password, devono essere protetti tramite hashing, le sessioni utente devono essere gestite in modo sicuro	Alta	Responsabile IT	Dati hashati, autenticazione attiva e funzionante
RNF-05	La piattaforma deve essere intuitiva e facile da usare, con UI semplice e funzionale per entrambi i profili (privati e aziende)	Alta	Utenti	Test di usabilità con $\geq 85\%$ di utenti che trovano l'interfaccia semplice e comprensibile

ID	Descrizione	Priorità	Fonte	Acceptance Criteria
RNF-06	Il sistema deve scalare per supportare un numero crescente di utenti senza degradare le performance o la reattività (implementazione futura)	Alta	Stakeholder	Test di carico mostrano performance stabili con aumento utenti
RNF-07	Compatibilità garantita con browser moderni (Chrome, Firefox, Safari) (implementazione futura)	Media	Utenti	Test di compatibilità sui principali browser
RNF-08	Codice sorgente deve essere ben documentato, con architettura modulare per facilitare manutenzione e futuri aggiornamenti	Media	Responsabile	Documentazione aggiornata e architettura modulare

Tabella 4: Requisiti Non Funzionali

I **requisiti di dominio** rappresentano vincoli e condizioni specifiche legate al contesto in cui il sistema viene sviluppato e utilizzato. Questi requisiti non definiscono direttamente le funzionalità del sistema, ma stabiliscono regole, normative, standard e limitazioni per garantire la conformità a leggi, policy aziendali e alle migliori pratiche del settore.

ID	Descrizione	Priorità	Fonte	Acceptance Criteria
RD-01	Il sistema deve garantire la conformità al GDPR per la protezione dei dati personali	Alta	Normativa Europea	Gestione dati conforme a GDPR
RD-02	CyberDefender deve essere conforme alla direttiva NIS2 per la sicurezza delle reti e dei sistemi informativi	Alta	Normativa Europea	Report di sicurezza periodici conformi a NIS2
RD-03	Il sistema deve implementare una gestione rigorosa di privilegi e accessi utente, con controllo basato sui ruoli	Alta	Policy Aziendale	Controllo accessi con ruoli definiti e verifica degli accessi
RD-04	Devono essere mantenuti log di auditing completi per tutte le operazioni critiche, garantendo la tracciabilità delle azioni	Media	Normativa e Best Practice	Log completi
RD-05	Il sistema deve rispettare standard internazionali di sicurezza, come ISO/IEC 27001	Media	Standard Internazionali	Certificazione o audit di conformità con standard ISO/IEC 27001
RD-05	Decreto Legislativo 196/2003	Alta	Legge Nazionale Italiana	Protezione dati personali

Tabella 5: Requisiti di Dominio

Personas – Profili Rappresentativi

Nel progetto, la definizione delle personas, ovvero profili rappresentativi degli utenti finali, è una fase cruciale per comprendere in modo approfondito le diverse esigenze, competenze e comportamenti degli utilizzatori del sistema. Le personas rappresentano quindi archetipi di utenti tipici, utili per simulare scenari d'uso realistici e per valutare l'efficacia delle soluzioni implementate dal punto di vista dell'esperienza utente (UX).

Per CyberDefender, sono state identificate due categorie principali di personas: l'utente privato, rappresentato dall’“utente classico”, e l'utente aziendale. Questo consente di sviluppare una piattaforma che risponda efficacemente alle esigenze di entrambe le figure.

Caratteristica	Laura Bianchi	Giulia Verdi
Nome	Laura Bianchi	Giulia Verdi
Età	45 anni	29 anni
Ruolo	Amministratore IT	Utente Privato
Professione	Responsabile sicurezza informatica	Impiegata amministrativa
Richiesta	<p>Come amministratrice IT di una media impresa Voglio avere una dashboard intuitiva con visualizzazione in tempo reale degli eventi di sicurezza e alert tempestivi In modo che possa monitorare la sicurezza aziendale, gestire le policy, e rispondere rapidamente a eventuali incidenti.</p>	<p>Come utente privato con competenze informatiche di base Voglio un’interfaccia semplice e notifiche comprensibili In modo che possa proteggere i miei dati personali senza complicazioni tecniche.</p>
Criteri di Accettazione	<ul style="list-style-type: none"> La dashboard mostra eventi di sicurezza aggiornati in tempo reale. Gli alert sono chiari e notificati tempestivamente. È possibile configurare e gestire i permessi degli utenti. Sono disponibili audit log dettagliati e accessibili. Il sistema supporta aggiornamenti automatici senza intervento manuale. Il sistema riduce al minimo i downtime durante aggiornamenti o manutenzioni. 	<ul style="list-style-type: none"> L’interfaccia è intuitiva e facile da usare. Le notifiche sono scritte in linguaggio semplice e chiaro. Il sistema offre protezione efficace contro phishing, malware e perdita dati. È disponibile un supporto rapido per chiarimenti e problemi. Le operazioni complesse sono automatizzate o guidate passo passo.
Timori	Minacce non rilevate, complessità, downtime	Perdita dati, phishing, malware invisibili

Tabella 6: Personas

Laura Bianchi, 45 anni, è l'amministratrice IT responsabile della sicurezza informatica in una media impresa. Utilizza CyberDefender quotidianamente per monitorare gli eventi di sicurezza, gestire le policy aziendali e rispondere prontamente a eventuali incidenti informatici. La sua priorità è disporre di una dashboard intuitiva che fornisca una visualizzazione in tempo reale degli eventi di sicurezza, oltre ad alert tempestivi che consentano interventi rapidi. Laura necessita di strumenti avanzati per la gestione dei permessi e per il mantenimento di audit log dettagliati, fondamentali per garantire la trasparenza e la conformità alle normative. Le sue principali preoccupazioni includono il rischio di minacce non rilevate, la complessità nella configurazione del sistema e possibili downtime che potrebbero compromettere la sicurezza aziendale. Inoltre, richiede funzionalità di automazione per gli aggiornamenti, al fine di mantenere sempre alta la protezione senza richiedere interventi manuali frequenti.

Giulia Verdi, 29 anni, è un'utente privata con competenze informatiche di base, che lavora come impiegata amministrativa. Usa CyberDefender principalmente per proteggere i propri dati personali e quelli della sua famiglia dalle minacce online. La sua esigenza principale è avere una soluzione semplice ed efficace, che le offra protezione senza complicazioni tecniche. Giulia desidera ricevere notifiche chiare e comprensibili, e un'interfaccia amichevole che la guida nell'uso della piattaforma. È particolarmente preoccupata dalla perdita di dati, da attacchi di phishing e da malware difficili da individuare. Per questo, ha bisogno di chiarimenti in linguaggio semplice, oltre a un supporto rapido in caso di problemi o dubbi.

Casi d'Uso

In questa sezione sono descritti i principali casi d'uso del sistema CyberDefender, con l'obiettivo di rappresentare le interazioni tra gli attori e le funzionalità offerte dalla piattaforma. Ogni caso d'uso illustra il comportamento atteso del sistema in risposta alle azioni degli utenti, evidenziando gli obiettivi funzionali, i flussi principali e le eventuali condizioni particolari.

Caso d'uso: Accesso Utente

Campo	Descrizione
ID	CU01
Titolo	Accesso Utente
Attore principale	Utente registrato
Stakeholder	Utente, Amministratore di sistema
Descrizione	L'utente inserisce username e password per accedere alla piattaforma. Il sistema verifica le credenziali e consente l'accesso solo se valide.
Precondizioni	L'utente deve essere registrato e avere credenziali valide.
Flusso principale	<ol style="list-style-type: none"> 1. L'utente inserisce username e password. 2. Il sistema autentica le credenziali. 3. Accesso consentito e visualizzazione dashboard. 4. Messaggio di errore in caso di credenziali errate.

Campo	Descrizione
Percorso Alternativo	Se l'utente dimentica la password, può utilizzare la funzione di recupero password tramite email.
Postcondizioni	L'utente è autenticato e può utilizzare le funzionalità della piattaforma secondo i propri permessi.
Eccezioni	Tentativi di accesso ripetuti con credenziali errate possono bloccare temporaneamente l'account (implementazione futura).
Requisiti associati	RF01 – Il sistema deve consentire il login con username e password RF02 – Il sistema deve permettere di impostare permessi di accesso alle aree private RF09 – Il sistema deve permettere il recupero della password tramite email RNF04 – Tutti i dati sensibili, come le password, devono essere protetti tramite hashing, le sessioni utente devono essere gestite in modo sicuro
Trigger	L'utente accede alla piattaforma e apre la schermata di login.
Frequenza d'uso	Quotidiana
Benefici per L'organizzazione	Accesso sicuro e tracciabile al sistema, controllo degli utenti attivi.

Tabella 7: Caso d'uso 1

Caso d'Uso: Consultazione minacce e misure difensive - Enciclopedia

Campo	Descrizione
ID	CU02
Titolo	Consultazione minacce e misure difensive - Enciclopedia
Attore principale	Utente
Stakeholder	Responsabile IT, Amministratore di sistema, Utente
Descrizione	L'utente accede alla sezione dedicata alle minacce rilevate e può visualizzare i dettagli degli attacchi, esempi e le misure difensive attivate.
Precondizioni	L'utente deve autorizzato a visualizzare gli eventi di sicurezza.
Flusso principale	1. L'utente accede alla dashboard sicurezza. 2. Seleziona la sezione delle Minacce. 3. Il sistema mostra l'enciclopedia delle minacce. 4. L'utente visualizza le specifiche della minaccia scelta.

Campo	Descrizione
Percorso Alternativo	-
Postcondizioni	L'utente ha ottenuto una panoramica delle minacce.
Eccezioni	Nessuna minaccia disponibile.
Requisiti associati	RF04 – Il sistema deve permettere la consultazione degli eventi di sicurezza RF07 – Auto-apprendimento per nuove minacce RNF03 – L'interfaccia utente deve essere chiara, accessibile e reattiva RNF05 – La piattaforma deve essere intuitiva e facile da usare, con UI semplice e funzionale per entrambi i profili (privati e aziende)
Trigger	L'utente seleziona "Minacce" dal menu della dashboard.
Frequenza d'uso	Frequente
Benefici per L'organizzazione	Monitoraggio e studio delle minacce e interventi rapidi.

Tabella 8: Caso d'uso 2

Caso d'uso: Analisi Input Sospetto (messaggi, numero telefonico)

Campo	Descrizione
ID	CU03
Titolo	Analisi di Input Sospetto
Attore principale	Utente registrato
Stakeholder	Utente
Descrizione	L'utente inserisce un input che il sistema analizza per valutarne la pericolosità.
Precondizioni	L'utente è autenticato nel sistema. L'input è in un formato supportato.
Flusso principale	<ol style="list-style-type: none"> 1. L'utente accede alla sezione "Analisi". 2. Inserisce l'input sospetto. 3. Clicca su "Analizza Messaggio o Numero Sospetto". 4. Il sistema esegue controlli. 5. Il risultato viene mostrato con indicazione del livello di rischio e suggerimenti.

Campo	Descrizione
Percorso Alternativo	Se l'input è troppo grande o in un formato non supportato, il sistema notifica l'errore.
Postcondizioni	L'utente visualizza un esito dell'analisi con eventuali azioni suggerite.
Eccezioni	L'utente non è loggato.
Requisiti associati	RF07 – Auto-apprendimento contro nuove minacce RF04 – Consultazione eventi di sicurezza RNF03 – L'interfaccia utente deve essere chiara, accessibile e reattiva RNF05 – La piattaforma deve essere intuitiva e facile da usare, con UI semplice e funzionale per entrambi i profili (privati e aziende)
Trigger	L'utente clicca “Analizza” nella sezione Input Sospetto.
Frequenza d'uso	Occasionale, usato in situazioni di dubbio
Benefici per L'organizzazione	Previene incidenti di sicurezza, aumenta consapevolezza e reattività degli utenti.

Tabella 9: Caso d'uso 3

Caso d'uso: Rilevamento Attacco

Campo	Descrizione
ID	CU04
Titolo	Rilevamento Attacco
Attore principale	Utente registrato
Stakeholder	Utente
Descrizione	L'utente accede alla sezione dedicata ai form di rilevamento attacchi, seleziona il form corrispondente, risponde alle domande e riceve il responso finale.
Precondizioni	L'utente ha effettuato il login ed è abilitato alla sezione del form.
Flusso principale	<ol style="list-style-type: none"> 1. L'utente accede alla sezione “Rilevamento Attacco”. 2. Completa tutte le domande proposte. 3. Invia il form. 4. Visualizza l'esito.
Percorso Alternativo	Se il form non viene completato, il sistema mostra un messaggio che invita l'utente a rispondere a tutte le domande prima dell'invio.

Campo	Descrizione
Postcondizioni	Il risultato viene salvato nel profilo utente.
Eccezioni	L'utente non è loggato.
Requisiti associati	RNF03 – Accessibilità e chiarezza UI RNF01 – Tempo di risposta inferiore a 3 secondi RNF05 – La piattaforma deve essere intuitiva e facile da usare, con UI semplice e funzionale per entrambi i profili (privati e aziende) RF07 – Auto-apprendimento contro nuove minacce
Trigger	L'utente accede alla sezione "Rilevamento Attacco".
Frequenza d'uso	Occasionale
Benefici per L'organizzazione	Migliora la cultura di sicurezza.

Tabella 10: Caso d'uso 4

Caso d'uso: Report in PDF

Campo	Descrizione
ID	CU05
Titolo	Report in PDF
Attore principale	Utente autorizzato
Stakeholder	Utente
Descrizione	L'utente scarica il report PDF della sua analisi di rilevamento attacco
Precondizioni	L'utente è autenticato e ha i permessi per accedere al rilevamento attacco.
Flusso principale	<ol style="list-style-type: none"> 1. L'utente compila il form. 2. Riceve l'esito. 3. Clicca sul pulsante per scaricare il pdf.
Percorso Alternativo	L'utente va nella sezione pdf e scarica il pdf.
Postcondizioni	Il file PDF è generato e disponibile per il download. L'utente può scaricarlo.
Eccezioni	- Permessi insufficienti → accesso negato alla funzione.
Requisiti associati	RF07 – Consultazione eventi di sicurezza

Campo	Descrizione
	RNF01 – Performance (generazione in \leq 3 secondi) RNF03 – L’interfaccia utente deve essere chiara, accessibile e reattiva RNF05 – La piattaforma deve essere intuitiva e facile da usare, con UI semplice e funzionale per entrambi i profili (privati e aziende) RF03 – Il download deve essere consentito solo se l’utente ha i permessi adeguati
Trigger	L’utente clicca sul pulsante per scaricare il pdf.
Frequenza d’uso	Occasionale
Benefici per L’organizzazione	Tracciabilità eventi, supporto alla documentazione per audit, valutazioni e conformità normativa.

Tabella 11: Caso d’uso 5

User Stories

Nel contesto dello sviluppo di una piattaforma di sicurezza informatica rivolta a utenti privati e aziendali, le User Stories rappresentano uno strumento fondamentale per garantire che le funzionalità implementate siano realmente centrate sulle esigenze degli utenti finali. L’approccio **user-centered**, infatti, permette di tradurre i requisiti funzionali e non funzionali in esperienze concrete, descrivendo ogni funzione dal punto di vista dell’utilizzatore.

Le User Stories raccolte in questo progetto derivano da un’analisi approfondita dei profili d’uso e mirano a coprire sia i bisogni operativi (come la consultazione di minacce, l’analisi di input sospetti o la verifica di numeri di telefono) sia quelli formativi e gestionali (rilevamento di attacchi, generazione di report PDF). Ogni storia si basa sulla formula “**Come [ruolo] voglio [obiettivo] così che [beneficio]**”, facilitando la comprensione trasversale tra analisti, sviluppatori e stakeholder aziendali. In particolare, le User Stories consentono di:

- Prioritizzare lo sviluppo in base al valore per l’utente e all’impatto sulla sicurezza.
- Garantire la tracciabilità dei requisiti, associando ciascuna storia ai casi d’uso e ai requisiti di sistema.
- Favorire il dialogo continuo tra team tecnico e referenti funzionali durante lo sviluppo iterativo e incrementale.

L’utilizzo di criteri di accettazione e l’associazione con i casi d’uso consente inoltre di mantenere allineata la progettazione con gli obiettivi di sicurezza, usabilità e performance.

ID	Titolo	Persona	User Story	Acceptance criteria
US01	Login Utente	Utente	Come utente voglio accedere alla piattaforma tramite login per poter utilizzare i servizi disponibili.	Deve essere possibile accedere solo se le credenziali dell’account sono corrette.

ID	Titolo	Persona	User Story	Acceptance criteria
US02	Recupero Password	Utente	Come utente voglio poter recuperare la password dimenticata tramite email per riottenere l'accesso al sistema.	L'utente può richiedere il recupero della password inserendo la propria email.
US03	Compilazione Form di Sicurezza	Utente	Come utente voglio poter compilare form di sicurezza per rilevare un attacco.	L'utente può accedere a un form che prevede campi obbligatori.
US04	Feedback Form	Utente	Come utente voglio ricevere un feedback dettagliato dopo un form per comprendere la tipologia di attacco.	Dopo la compilazione del form, l'utente riceve un feedback dettagliato.
US05	Report PDF	Utente	Come utente voglio scaricare un report PDF dei miei form	Il PDF deve contenere data, informazioni dell'utente e dati del form.
US06	Analisi Messaggio Sospetto	Utente	Come utente voglio poter inserire un messaggio sospetto per analizzarne il livello di rischio.	Il processo di analisi è completato entro un tempo ragionevole (es. pochi secondi).
US07	Visualizzazione Risultato Analisi	Utente	Come utente voglio vedere i risultati dell'analisi in modo chiaro per capire se l'input è sicuro o dannoso.	I risultati dell'analisi sono mostrati in modo chiaro e leggibile.
US08	Verifica Numero di Telefono	Utente	Come utente voglio inserire un numero sospetto per sapere se è pericoloso.	L'utente può inserire un numero di telefono sospetto ed eseguire l'analisi.
US09	Encyclopedia	Utente, Amministratore	Come utente voglio consultare un elenco aggiornato delle minacce rilevate per essere consapevole dei rischi.	L'elenco è facilmente consultabile e filtrabile per categoria.
US10	Visualizzazione Misure Difensive	Utente	Come utente voglio vedere le contromisure per una specifica minaccia così da comprendere il tipo di protezione attiva.	L'utente può vedere le contromisure attive per una specifica minaccia.

Tabella 12: User Stories

Matrice di Tracciabilità dei Requisiti

La matrice di tracciabilità dei requisiti rappresenta uno strumento essenziale per assicurare che tutti i requisiti identificati nel progetto CyberDefender vengano correttamente implementati, verificati e mantenuti nel corso del tempo. Questa matrice collega in modo sistematico i requisiti funzionali, non funzionali e di dominio con i principali artefatti del progetto, quali user stories, casi

d'uso, test case e componenti di sistema. Grazie a questa correlazione, la matrice garantisce un elevato livello di qualità, coerenza e facilità di manutenzione del sistema di sicurezza CyberDefender, facilitando inoltre la gestione e il controllo dell'intero processo di sviluppo.

ID	Ass. ID	Requirements Description	Business Need / Justification	Project Objective	Requested By	Department	Specification	Test Cases	Stato
RF-01	CU01, US01	Login con username e password	Garantire accesso sicuro e controllato alla piattaforma	Abilitare autenticazione utenti	Utente	IT Security	Login, verifica credenziali	TC-01: Login valido/errato	Implementato
RF-02	CU01	Impostare permessi di accesso	Controllo degli accessi a sezioni riservate	Applicare il modello RBAC (controllo accessi)	Utente	IT Security	Gestione ruoli e permessi	TC-02: Accesso aree secondo ruolo	Implementato
RF-03	CU05	Download consentito solo se autorizzato	Protezione documenti sensibili	Controllare l'accesso al download PDF	Utente	IT Security	Verifica permessi prima del download	TC-03: Accesso consentito/ negato	Implementato
RF-04	CU02, CU03	Consultazione eventi di sicurezza	Rilevazione tentativi di intrusione	Fornire visibilità delle minacce	Responsabile IT	IT Security	Dashboard eventi	TC-04: Visualizzazione tentativi	Implementato
RF-06	-	Scalabilità su migliaia di dispositivi	Supportare ambienti enterprise	Ottimizzare performance su larga scala	Stakeholder	Architettura	Test con ≥ 100 dispositivi (implementazione futura)	TC-06: Carico >1000 device	Pianificato
RF-09	CU01	Recupero password tramite email	Migliorare UX, evitare blocchi utenti	Facilitare riaccesso	Utente	Supporto utenti	Funzione "Password dimenticata" (implementazione futura)	TC-09: Email inviata / reset completato	Pianificato
RNF-01	CU03 – CU05	Tempo risposta ≤ 3 sec	Esperienza utente fluida	Ottimizzare tempo di risposta	Stakeholder	UX / IT Performance	Test temporale operazioni comuni	TC-10: Tempo < 3 s su funzioni chiave	Implementato

ID	Ass. ID	Requirements Description	Business Need / Justification	Project Objective	Requested By	Department	Specification	Test Cases	Stato
RNF-03	Tutti	UI chiara, accessibile e reattiva	Usabilità inclusiva	Rendere piattaforma user-friendly	Utenti	UX	Interfaccia semplice e responsive	TC-11: Feedback usabilità	Implementato
RNF-04	CU01	Dati sensibili protetti (hashing, sessioni sicure)	Conformità e sicurezza dei dati	Protezione dati utente	Resp. IT	IT Security	Hashing, sessioni	TC-12: Verifica sicurezza auth	Implementato

Tabella 13: Matrice tracciabilità dei Requisiti

Questa tabella consente di mantenere l'allineamento tra requisiti, sviluppo, testing e obiettivi degli utenti. I campi dedicati ai "Test Case" sono fondamentali per il collaudo finale e per la documentazione della qualità (QA). Le "User Story" associate aiutano a comprendere chi sono gli utenti coinvolti, quali sono le loro esigenze e il motivo dietro ogni requisito.

Misuse Case: Rischi e Contromisure

Nel contesto del progetto CyberDefender, è essenziale non solo analizzare i casi d'uso positivi ma anche identificare e gestire i misuse case, ovvero quegli scenari in cui utenti malintenzionati o condizioni anomale potrebbero compromettere la sicurezza, l'integrità o la disponibilità del sistema. Questa sezione si concentra sui rischi potenziali derivanti dall'uso improprio della piattaforma o da tentativi di attacco informatico, descrivendo le azioni indesiderate che potrebbero essere compiute e le contromisure da adottare per mitigare efficacemente.

Attraverso l'analisi dettagliata dei misuse case, CyberDefender è progettato per prevenire, rilevare e rispondere in modo tempestivo e adeguato a minacce quali accessi non autorizzati, manipolazione dei dati, attacchi DDoS, distribuzione di malware e altre forme di abuso. Questo garantisce un elevato livello di sicurezza per gli utenti e per le infrastrutture aziendali. Le principali minacce identificate includono:

- **Manipulate Logs:** attività di utenti malintenzionati mirate a modificare o cancellare i log di sistema per nascondere azioni sospette o non autorizzate, riducendo l'efficacia di audit e tracciabilità.
- **Manipulate URL:** alterazione di URL in messaggi o comunicazioni per indirizzare gli utenti verso siti malevoli (phishing), bypassando i controlli di sicurezza.
- **Brute-force Attack:** tentativi sistematici di indovinare credenziali di accesso al sistema con l'obiettivo di ottenere accessi non autorizzati.
- **Denial of Service (DoS):** generazione di traffico malevolo o richieste massicce per sovraccaricare la piattaforma, rendendola indisponibile agli utenti legittimi e compromettendo la continuità del servizio.

- **Privilege Escalation:** tentativi da parte di utenti con permessi limitati di ottenere privilegi più elevati sfruttando vulnerabilità o errori di configurazione, per manipolare o controllare funzionalità riservate.

Le minacce e le contromisure sono mappate secondo il modello STRIDE, che assicura una copertura completa delle categorie chiave di attacchi:

- **Spoofing** (Falsificazione): prevenzione di accessi non autorizzati tramite meccanismi di autenticazione sicura e crittografia delle sessioni.
- **Tampering** (Manomissione): protezione contro modifiche non autorizzate di file e messaggi, grazie all'utilizzo di firme digitali e controlli di integrità.
- **Repudiation** (Negazione delle azioni): implementazione di log dettagliati e tracciabilità per impedire la negazione di operazioni da parte degli utenti.
- **Information Disclosure** (Divulgazione di informazioni): crittografia end-to-end dei dati sensibili per prevenire fughe di informazioni.
- **Denial of Service** (Interruzione del servizio): sistemi di monitoraggio e difesa per garantire la disponibilità continua della piattaforma.
- **Elevation of Privilege** (Elevazione dei privilegi): rigidi controlli di accesso e gestione dei ruoli per impedire acquisizioni non autorizzate di permessi elevati.

Questa strategia integrata consente a CyberDefender di offrire una piattaforma sicura, affidabile e conforme alle normative vigenti per la protezione dei dati e la gestione efficace delle minacce informatiche.

Come CyberDefender protegge dalle injection attacks:

- **Validazione e sanificazione degli input:** Ogni dato ricevuto dall'utente (messaggi, nomi file, parametri) viene rigorosamente controllato e pulito per rimuovere caratteri o comandi potenzialmente pericolosi, prevenendo l'esecuzione di codice malevolo.
- **Uso di query parametrizzate e prepared statements:** Nelle operazioni con database, vengono utilizzate query che separano dati da codice, impedendo l'esecuzione di comandi iniettati.
- **Controlli di integrità e autenticazione:** Verifica dell'integrità dei dati e applicazione di autenticazioni forti per limitare l'accesso a utenti autorizzati.
- **Monitoraggio e rilevamento anomalie:** Controllo continuo del traffico e delle attività per identificare e bloccare tempestivamente tentativi di injection.

ID	Misuse Case	Minaccia (STRIDE)	Attore Malevolo	Descrizione	Conseguenze	Contromisure
MU01	Accesso non autorizzato	Spoofing	Hacker esterno	Un attaccante tenta di impersonare un utente legittimo per accedere al sistema.	Accesso a dati sensibili, violazione privacy	Hashing Argon2, sessioni
MU02	Manomissione del report PDF	Tampering	Utente interno malevolo	L'utente modifica un report PDF generato per	Perdita di integrità, inganno di altri utenti o stakeholder	Firma digitale sui report (implementazione futura)

ID	Misuse Case	Minaccia (STRIDE)	Attore Malevolo	Descrizione	Conseguenze	Contromisure
				nascondere informazioni critiche.		
MU03	Furto di credenziali	Information Disclosure	Malware, Phishing	L'attaccante ruba le credenziali tramite phishing o keylogger.	Accesso completo al sistema, blocco account, perdita dati	Protezione da phishing, educazione dell'utente, login sicuro, monitoraggio
MU04	Invio massivo di richieste alla dashboard	Denial of Service	Botnet	Un attacco DoS rende la dashboard inaccessibile agli utenti legittimi.	Interruzione dei servizi, rallentamento, blocco amministratori	Bilanciamento carico (implementazione futura)
MU05	Accesso a eventi riservati	Elevation of Privilege	Utente con permessi limitati	Un utente normale accede ad aree amministrative non autorizzate.	Violazione integrità configurazioni, danni al sistema	Controllo accessi per ruolo (RBAC), logging accessi, revisione permessi
MU06	Visualizzazione log da parte di utenti base	Information Disclosure	Utente non privilegiato	Utente visualizza audit log senza autorizzazione.	Esposizione informazioni interne e dettagli su attacchi	Mascheramento dati, separazione ruoli, restrizione accesso (implementazione futura)

Tabella 14: Misuse Cases

Requisiti di privacy e compliance

Nel contesto della sicurezza informatica, la protezione dei dati personali e la conformità alle normative vigenti rappresentano una componente essenziale per il corretto funzionamento della piattaforma CyberDefender. Questa sezione descrive i requisiti di privacy e le misure di compliance che il sistema deve rispettare per garantire l'integrità, la riservatezza e la disponibilità delle informazioni trattate, in linea con i principali riferimenti normativi nazionali e internazionali. Bisogna quindi:

- Garantire la conformità al **Regolamento Generale sulla Protezione dei Dati (GDPR)**.
- Ridurre i rischi legati al trattamento illecito o non autorizzato dei dati.
- Fornire trasparenza, tracciabilità e controllo sui dati degli utenti.
- Rispondere alle esigenze di audit, accountability e reporting richieste dalle autorità competenti.

Requisiti di Privacy

Requisito	Descrizione
Minimizzazione dei dati	Il sistema raccoglie solo i dati strettamente necessari al funzionamento (principio di minimizzazione).
Diritto all'Oblio	Il sistema implementa procedure automatizzate per la cancellazione dei dati su richiesta dell'utente (implementazione futura).
Accesso controllato	I dati personali sono accessibili solo da utenti autorizzati tramite controllo di accesso.
Log delle attività	Tutti gli accessi e le attività sono monitorate.

Tabella 15: Requisiti di Privacy

Requisiti Normativi

Riferimento Normativo	Requisito	Applicazione in CyberDefender
GDPR Art. 5-6	Trattamento lecito, trasparente e limitato alle finalità	Privacy policy chiara, logging dettagliato, mascheramento dati personali
GDPR Art. 32	Sicurezza del trattamento	Implementazione di misure tecniche e organizzative (crittografia, hashing, backup, accesso controllato)
ISO/IEC 27001	Sistema di Gestione della Sicurezza delle Informazioni	Politiche di sicurezza, analisi dei rischi, piani di risposta agli incidenti
Direttiva NIS2 (UE)	Resilienza e sicurezza dei sistemi informativi critici	Continuità operativa, report automatici verso autorità di vigilanza, gestione incidenti (implementazione futura)
Legge 231/2001 (Italia)	Responsabilità amministrativa delle imprese	Logging, monitoraggio accessi, audit trail accessibili per verifiche
Codice Privacy (D.Lgs. 196/2003)	Protezione dati personali	Integrazione con registro dei trattamenti e DPIA (valutazione d'impatto)
OWASP Top 10 Compliance	Prevenzione vulnerabilità note	Controllo input, protezione contro injection, XSS, session hijacking

Tabella 16: Requisiti Normativi

CyberDefender adotta un insieme strutturato di misure tecniche e organizzative per garantire la conformità normativa e la protezione dei dati:

- **Diritto all'oblio:** Il sistema prevede procedure semplificate per la cancellazione totale dei dati personali, in conformità all'Art. 17 del GDPR.
- **Sicurezza e minimizzazione:** Viene applicato il principio di minimizzazione, raccogliendo solo i dati strettamente necessari, gestiti con hashing e controllo degli accessi.

- **Compliance con NIS2:** La piattaforma è progettata per assicurare l'operatività continua, la gestione tempestiva degli incidenti.
- **Standard internazionali (NIST e ISO):** CyberDefender si ispira a framework come il NIST SP 800-53, il NIST CSF e lo standard ISO/IEC 27001, per rafforzare la gestione del rischio, la sicurezza operativa e la resilienza infrastrutturale.

4. Analisi delle minacce (Threat Modeling)

Il Threat Modeling è un'attività fondamentale per la sicurezza, caratterizzata da un approccio pragmatico piuttosto che rigidamente formalizzato. Il suo obiettivo principale è identificare e correggere potenziali problemi di sicurezza il prima possibile, preferibilmente già durante la fase di progettazione o sviluppo del sistema. Comprendere in profondità i requisiti di sicurezza permette non solo di prevenire bug critici, ma anche di migliorare la qualità ingegneristica complessiva del prodotto.

Obiettivo e metodo

L'analisi delle minacce rappresenta una fase cruciale nel processo di sicurezza della piattaforma CyberDefender, volta a individuare e valutare in modo sistematico tutte le potenziali vulnerabilità e scenari di attacco che potrebbero compromettere l'integrità, la riservatezza e la disponibilità del sistema. Questo approccio permette di anticipare possibili vettori di attacco e di predisporre adeguate contromisure per proteggere sia i dati sensibili degli utenti sia la corretta funzionalità della piattaforma, migliorandone la robustezza e la resilienza complessiva.

Per raggiungere questi obiettivi, si adotta un metodo strutturato di Threat Modeling, basato principalmente sul modello STRIDE (Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, Elevation of Privilege). Questo modello facilita l'identificazione di specifiche categorie di minacce che possono colpire i diversi componenti e flussi informativi della piattaforma.

L'analisi parte da una dettagliata mappatura dei flussi informativi mediante la costruzione di Data Flow Diagram (DFD), che rappresentano graficamente le interazioni tra utenti, processi, dati e componenti di sistema. Questo consente di visualizzare chiaramente i punti critici e i vettori potenziali di attacco.

Per ciascun elemento del DFD, ovvero ogni processo, datastore, attore esterno e flusso di dati, vengono identificate le minacce rilevanti secondo la **classificazione STRIDE**. Successivamente, si procede con una valutazione del rischio associato ad ogni minaccia, considerando fattori come la probabilità di sfruttamento e l'impatto potenziale sull'ecosistema CyberDefender.

Infine, per ogni minaccia individuata, vengono definite e documentate **strategie di mitigazione mirate**. Queste possono includere misure tecniche (come l'implementazione di controlli di accesso, crittografia, validazione degli input), organizzative (procedure di monitoraggio e risposta agli incidenti) e di design (architettura di sistema sicura e ridondante).

L'intero processo è iterativo e integrato nel ciclo di sviluppo software, garantendo che la sicurezza sia un elemento costante e prioritario sin dalle prime fasi progettuali fino al rilascio e alla manutenzione della piattaforma.

Data Flow Diagram (DFD) - level 0

I Data Flow Diagram (DFD) di livello 0 rappresentano uno strumento particolarmente efficace per la modellazione delle minacce, in quanto consentono di analizzare il sistema seguendo il flusso dei dati, che è spesso il vettore principale di problemi di sicurezza, più del semplice flusso di controllo. Questi diagrammi permettono di identificare e visualizzare in modo chiaro gli elementi fondamentali del sistema: le entità esterne, i processi, i flussi di dati, gli archivi di dati e i trust boundaries — ovvero

i confini che separano zone con diversi livelli di fiducia. Tale rappresentazione facilita l'individuazione di potenziali punti critici, contribuendo a una valutazione più mirata ed efficace delle minacce.

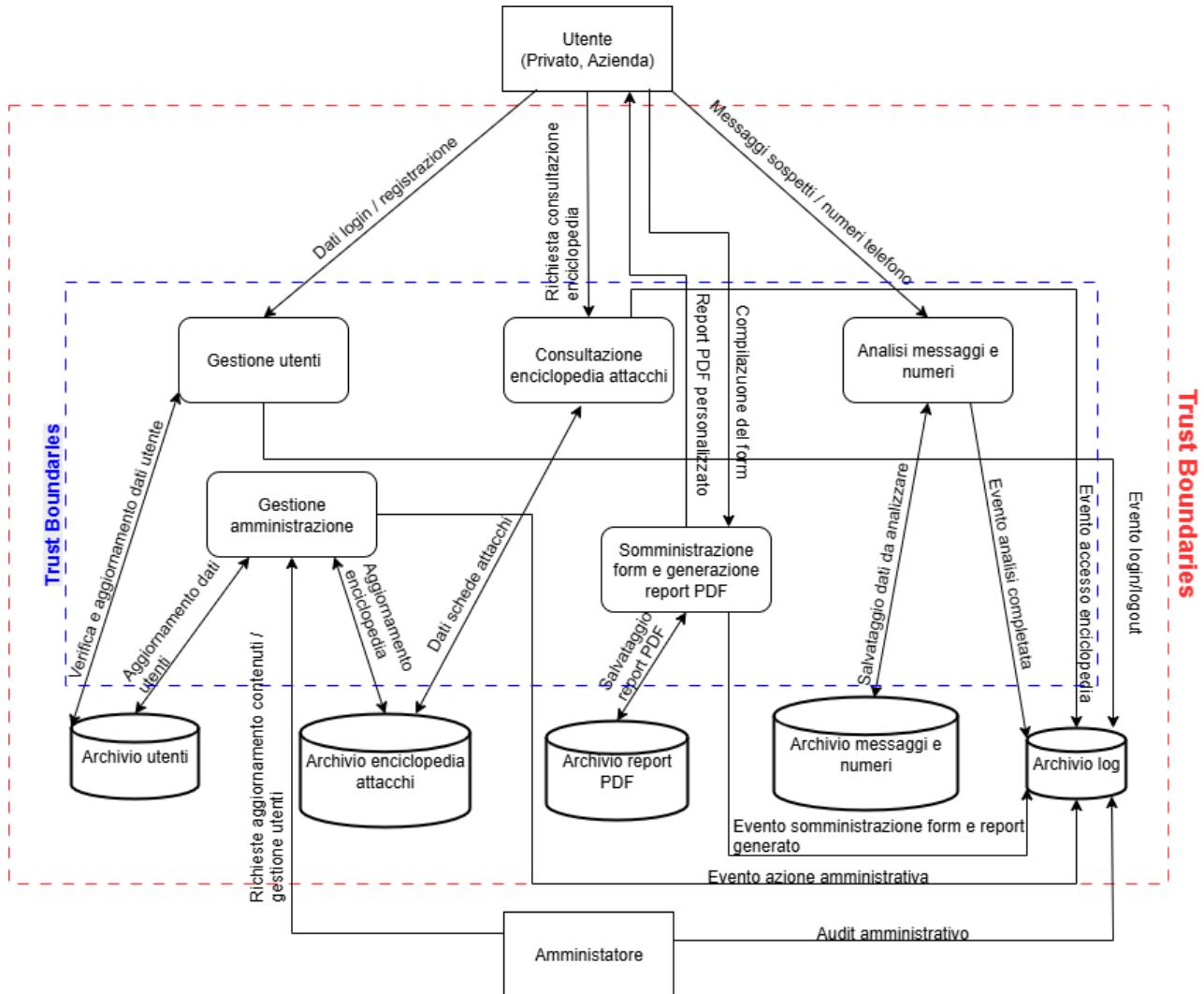


Figura 1: Data Flow Diagram (DFD) livello 0: interazioni tra entità esterne (utente e amministratore), i flussi di dati, gli archivi e i confini di fiducia (trust boundaries).

Legenda colori

Forma	Significato
Rettangolo	Entità Esterne
Rettangolo arrotondato	Processo
Cilindro	Archivio dati
Frecce	Flusso dati
Rettangolo tratteggiato	Trust Boundary

Tabella 17: Legenda Colori

Le componenti presenti nella Figura 1, sono le seguenti:

Entità Esterne

Le entità esterne rappresentano gli attori che interagiscono con il sistema ma non fanno parte di esso. Nel caso di CyberDefender, le entità esterne sono:

- **Utente**: può essere un privato o un'azienda. Accede alle funzionalità del sistema per consultare l'enciclopedia degli attacchi, analizzare messaggi sospetti o numeri di telefono, eseguire test di valutazione e ricevere report personalizzati.
- **Amministratore**: gestisce i contenuti della piattaforma, supervisiona le attività del sistema e può accedere a log e dati utente per scopi di auditing e manutenzione.

Processi

I **processi** rappresentano le trasformazioni dei dati nel sistema. Ogni processo riceve input da entità esterne o archivi, esegue un'elaborazione e restituisce output. I principali processi nel DFD di *CyberDefender* sono:

- **P1 – Gestione utenti**: registra nuovi utenti, verifica credenziali di accesso, assegna permessi e scrive eventi nei log.
- **P2 – Consultazione enciclopedia**: permette all'utente di accedere alle schede informative sugli attacchi informatici.
- **P3 – Analisi messaggi e numeri**: analizza contenuti sospetti (email, testo o numeri di telefono) inviati dall'utente per identificare minacce.
- **P4 – Somministrazione form e generazione report PDF**: somministra un form e genera un documento personalizzato in base ai risultati dell'analisi dei form.
- **P5 – Gestione amministrativa**: consente all'amministratore di aggiornare i contenuti dell'enciclopedia, consultare log di sistema e gestire gli utenti.

Flussi di dati

I **flussi di dati** indicano il passaggio di informazioni tra entità, processi e archivi. Ogni flusso rappresenta un'informazione significativa per il funzionamento del sistema. Alcuni esempi:

- “Dati registrazione” → da Utente a P1
- “Messaggio da analizzare” → da Utente a P3
- “Report PDF” → da P4 a Utente
- “Aggiornamento contenuti” → da Amministratore a P5
- “Evento di log” → da ogni processo verso l’Archivio log (D5)

I flussi sono rappresentati nel diagramma con **frecce orientate**, etichettate con il tipo di dato trasferito.

Archivi di dati

Gli **archivi di dati** (o data store) rappresentano le basi di dati statiche in cui il sistema conserva informazioni. Nel sistema CyberDefender, gli archivi principali sono:

- **D1 – Archivio utenti:** contiene dati personali, credenziali cifrate e permessi degli utenti.
- **D2 – Archivio enciclopedia attacchi:** raccoglie le schede informative sugli attacchi informatici.
- **D3 – Archivio messaggi e numeri:** memorizza messaggi e numeri inviati dagli utenti per l’analisi.
- **D4 – Archivio report PDF:** conserva i report personalizzati generati dal sistema dopo l’analisi dei risultati dei form.
- **D5 – Archivio log:** registra tutte le attività rilevanti del sistema, come login, analisi, modifiche da parte dell’amministratore, ecc

Confini di fiducia (Trust Boundaries)

I **confini di fiducia** indicano le separazioni tra aree con livelli di sicurezza differenti. Servono a evidenziare dove i dati attraversano zone con diverso controllo di accesso. Nel caso di *CyberDefender* si distinguono:

- **Trust boundary tra Utente e Sistema (linea rossa tratteggiata superiore):** rappresenta l’interfaccia tra un utente non autenticato/autenticato e le funzionalità del sistema. Qui avvengono controlli su credenziali, input sospetti e permessi.
- **Trust boundary tra Amministratore e Sistema (linea rossa tratteggiata inferiore):** l’amministratore ha accesso privilegiato a funzioni sensibili. Questo confine separa la zona ad alto privilegio dal resto del sistema.
- **Trust boundary interno tra Processi e Archivi (linea blu tratteggiata):** Indica la separazione tra i processi attivi del sistema (es. analisi, gestione, consultazione) e i dati persistenti memorizzati negli archivi:
 - Archivio utenti
 - Archivio enciclopedia attacchi
 - Archivio messaggi e numeri
 - Archivio report PDF
 - Archivio log

Questo confine evidenzia che l’accesso agli archivi avviene attraverso meccanismi controllati e monitorati, e può essere soggetto a restrizioni differenti rispetto all’esecuzione dei processi stessi.

DFD level 1 - Sprint 1 – Consultazione dell'enciclopedia attacchi

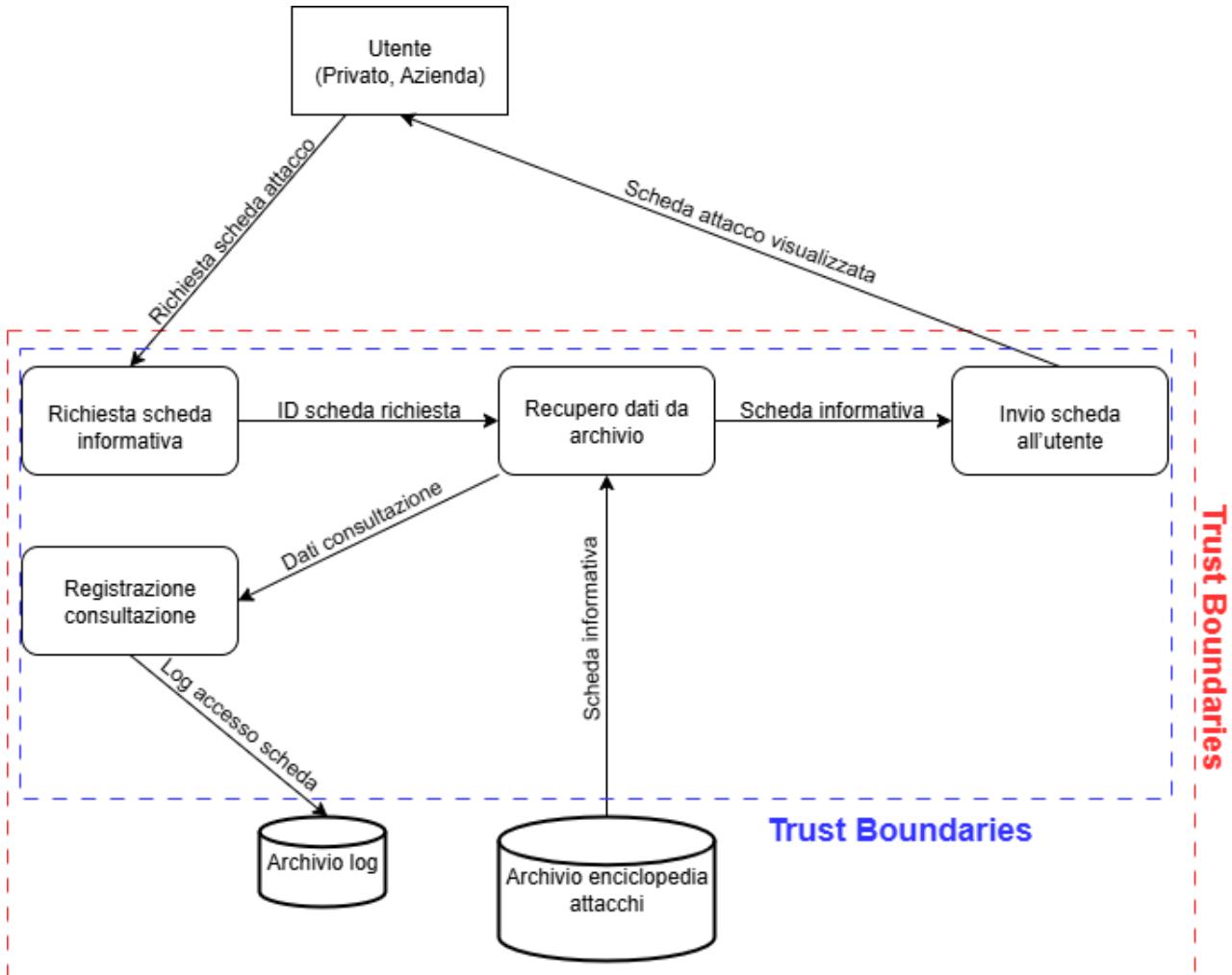


Figura 2: DFD Livello 1: Consultazione enciclopedia attacchi

Quando un utente vuole consultare informazioni sugli attacchi informatici, inizia inviando una richiesta specifica al sistema, selezionando la scheda che desidera visualizzare. Il sistema riceve questa richiesta e si prepara a recuperare i dati necessari.

Successivamente, il sistema utilizza l'identificativo della scheda richiesta per cercare all'interno dell'archivio enciclopedia le informazioni dettagliate sull'attacco selezionato. Questa ricerca restituisce i contenuti completi, tra cui la definizione, le modalità di esecuzione, le conseguenze e i consigli pratici per la prevenzione.

Per garantire la sicurezza e la tracciabilità, il sistema registra l'evento di consultazione nei propri log, includendo dati come l'identificativo dell'utente (se disponibile), il momento esatto della consultazione e quale scheda è stata visualizzata. Questo passaggio è importante per eventuali audit futuri.

Infine, il sistema invia i dati recuperati all'interfaccia utente, permettendo così all'utente di visualizzare la scheda informativa completa sull'attacco informatico richiesto. A questo punto, la consultazione si considera conclusa.

Come indicato nel diagramma, le interazioni con l'utente avvengono oltre il confine di fiducia, mentre i processi interni, gli archivi e il modulo di logging operano in un'area controllata dal sistema.

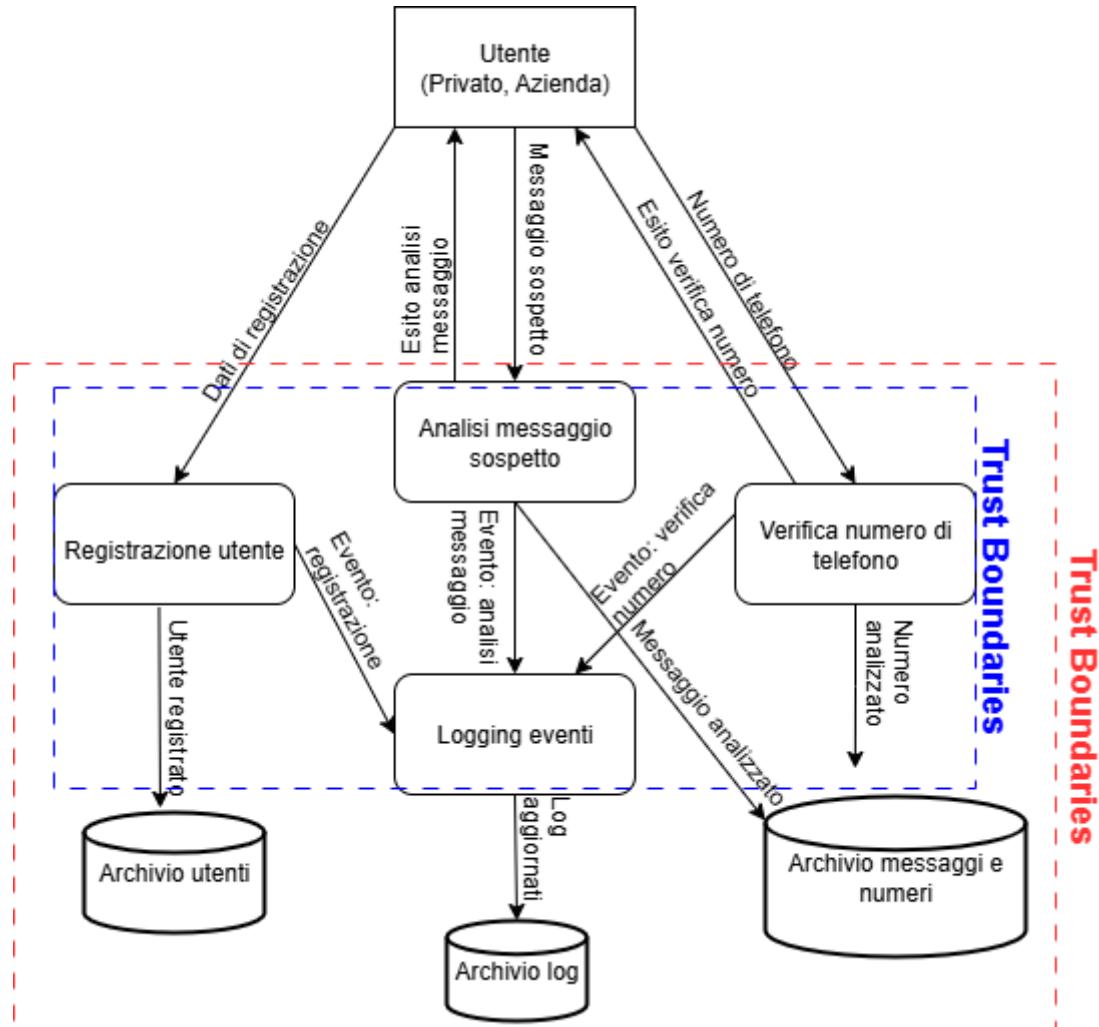


Figura 3: Implementazione login e analisi natura messaggi e numeri

Nel secondo sprint, il sistema CyberDefender introduce tre funzionalità fondamentali: la registrazione degli utenti, l’analisi dei messaggi sospetti e la verifica dei numeri di telefono. Questi processi condividono una struttura coerente e un archivio comune, ottimizzando l’organizzazione dei dati.

La registrazione utente prevede che l’utente inserisca le proprie credenziali. Il sistema valida i dati, cifra la password e li salva in un archivio dedicato. Viene anche generato un log per registrare l’evento.

L’analisi di messaggi sospetti consente all’utente di inviare contenuti sospetti (come email o messaggi di testo). Il sistema li elabora per identificare pattern dannosi. I risultati vengono salvati in un archivio condiviso con le verifiche telefoniche, e anche questo evento viene registrato nei log.

Con la verifica dei numeri di telefono, l’utente può controllare numeri sospetti. Il sistema risponde con un giudizio sull’affidabilità del numero e ne registra l’input nello stesso archivio dei messaggi, mantenendo centralizzata la gestione dei dati analizzati.

Un unico archivio, chiamato Archivio messaggi e numeri, contiene quindi sia i messaggi sospetti sia i numeri telefonici verificati. Tutte le attività vengono monitorate e registrate nel sistema di log, che garantisce tracciabilità e sicurezza.

Come indicato nel diagramma, le interazioni con l'utente avvengono oltre il confine di fiducia, mentre i processi interni, gli archivi e il modulo di logging operano in un'area controllata dal sistema.

DFD level 1 - Sprint 3 – Implementazione form e report PDF

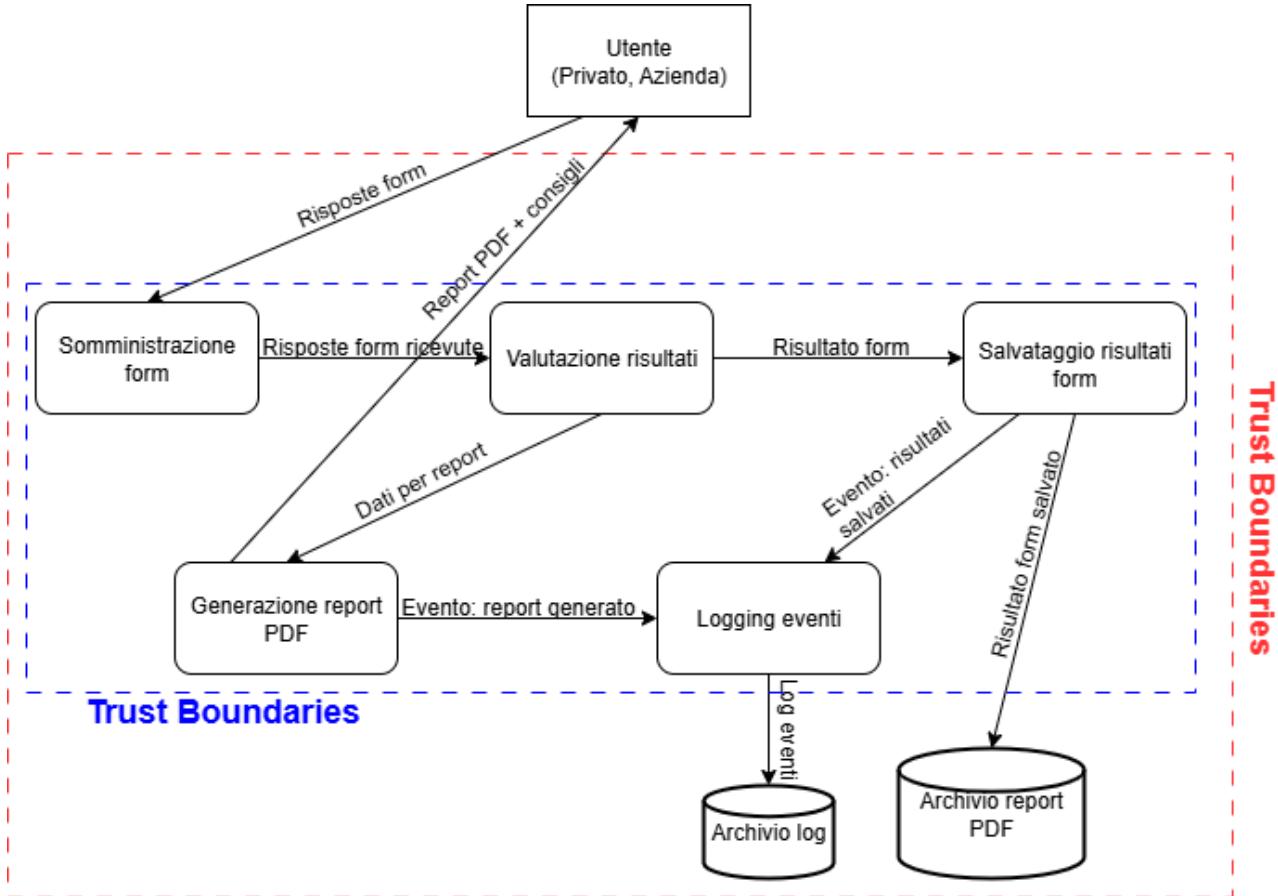


Figura 4: DFD di livello 1: Implementazione form e generazione del report PDF

La figura mostra il Data Flow Diagram di livello 1 relativo alla funzionalità implementata nello Sprint 3, che comprende la somministrazione di un form all'utente, il salvataggio dei risultati, e la generazione automatica di un report in formato PDF con consigli personalizzati.

L'utente, esterno al sistema, invia le risposte del form tramite l'interfaccia. Queste vengono gestite dal processo di somministrazione del form, che le inoltra al componente di valutazione dei risultati. Quest'ultimo analizza le risposte, calcola un punteggio e genera una valutazione finale.

I risultati ottenuti vengono quindi inviati al componente di salvataggio, che li registra all'interno dell'archivio dei risultati del form. In parallelo, un processo di logging registra l'evento nel sistema di audit (archivio dei log).

Successivamente, la valutazione prodotta viene utilizzata per creare un report PDF contenente consigli personalizzati. Il report viene restituito all'utente, mentre l'evento di generazione viene anch'esso registrato a fini di tracciabilità.

Tutti i processi sono contenuti in un trust boundary e gli archivi sono contenuti in un altro trust boundary. L'utente, invece, è considerato esterno e si trova quindi al di fuori di queste due aree.

STRIDE – Classificazione delle minacce

STRIDE è un modello per classificare le minacce alla sicurezza in sei categorie. Aiuta a identificare i diversi tipi di attacchi che possono compromettere un sistema, facilitando l'analisi dei rischi e la definizione di contromisure.

Categoria	Descrizione	Esempio
Spoofing	Falsificazione dell'identità di un'entità, utente o dispositivo.	Un attaccante accede a un sistema fingendosi un utente legittimo.
Tampering	Manomissione o modifica non autorizzata di dati, codice o configurazioni.	Un file di configurazione viene alterato per disattivare i controlli di sicurezza.
Repudiation	Negazione di un'azione senza possibilità di tracciarla o dimostrarla.	Un utente nega di aver effettuato una transazione perché non è stata registrata.
Information Disclosure	Esposizione non autorizzata di dati sensibili o riservati.	Un sito web mostra dati personali di altri utenti a causa di un errore.
Denial of Service	Interruzione o degrado del servizio, impedendo l'accesso agli utenti legittimi.	Un attacco DDoS manda in crash un sito web, rendendolo inaccessibile.
Elevation of Privilege	Un utente ottiene accessi o privilegi superiori a quelli autorizzati.	Un normale utente sfrutta una vulnerabilità per ottenere diritti da amministratore.

Tabella 18: Classificazione delle minacce STRIDE

Minacce specifiche identificate

In questa sezione vengono presentate le minacce individuate sui principali componenti del sistema CyberDefender, classificate secondo il modello STRIDE.

Per ciascuna minaccia, sono fornite:

- Una descrizione dettagliata
- Le modalità di mitigazione adottate
- Lo sprint nel quale sono state implementate

1. Interfaccia utente (UI)

Minaccia (STRIDE)	Descrizione	Gravità	Sprint	Mitigazione
Spoofing	Attaccante impersona un utente legittimo	Alta	2	Autenticazione forte, monitoraggio accessi sospetti
Tampering	Modifica del contenuto visualizzato	Media	1	Validazione input
Information Disclosure	Esposizione accidentale di dati sensibili tramite UI	Alta	2	Crittografia Argon2, accessi differenziati, mascheramento dati sensibili

Tabella 19: Minaccia interfaccia utente

2. Modulo di autenticazione

Minaccia (STRIDE)	Descrizione	Gravità	Sprint	Mitigazione
Spoofing	Furto delle credenziali o session hijacking	Alta	2	Gestione sicura delle sessioni
Repudiation	Negazione di login o azioni fatte	Media	2	Log dettagliati
Elevation of Privilege	Accesso a funzioni admin con credenziali rubate	Alta	2	Controlli di accesso

Tabella 20: Minaccia Modulo Autenticazione

3. Database

Minaccia (STRIDE)	Descrizione	Gravità	Sprint	Mitigazione
Tampering	Modifica non autorizzata dei dati	Alta	1	Controlli di integrità, accessi limitati
Information Disclosure	Accesso non autorizzato a dati sensibili	Alta	1	Crittografia dati a riposo, controllo degli accessi

Minaccia (STRIDE)	Descrizione	Gravità	Sprint	Mitigazione
Denial of Service	Saturazione delle risorse del database	Media	1	Limiti di query, monitoraggio attività anomala

Tabella 21: Minaccia Database

4. Server applicativo

Minaccia (STRIDE)	Descrizione	Gravità	Sprint	Mitigazione
Tampering	Inserimento di codice malevolo	Alta	2	Code review
Elevation of Privilege	Sfruttamento di vulnerabilità per aumentare privilegi	Alta	2	Controllo degli accessi
Denial of Service	Attacco volto a saturare le risorse del server	Media	3	Rate limiting, firewall, monitoraggio della rete (implementazione futura)

Tabella 22: Minaccia Server

5. Sistema di logging e monitoraggio

Minaccia (STRIDE)	Descrizione	Gravità	Sprint	Mitigazione
Repudiation	Mancata registrazione o modifica dei log	Alta	2	Log immutabili
Tampering	Alterazione dei log per nascondere attività malevole	Alta	2	Protezione dei log, monitoraggio integrità
Denial of Service	Sovraccarico del sistema di logging	Media	2	Scalabilità del sistema di logging

Tabella 23: Minaccia logging

Questa analisi consente di comprendere i rischi specifici e le contromisure implementate per garantire la sicurezza del sistema.

Misuse Case rilevanti

Nel contesto del progetto CyberDefender, è fondamentale non solo analizzare i casi d'uso positivi, ma anche identificare e gestire i misuse case: scenari in cui utenti malintenzionati o condizioni anomale potrebbero compromettere la sicurezza, l'integrità o la disponibilità del sistema.

ID	Misuse Case	Minaccia (STRIDE)	Attore Malevolo	Obiettivo	Contromisure
MU01	Manipolazione Log di Sistema	Repudiation, Tampering	Utente malintenzionato con accesso ai log	Modificare o cancellare log per nascondere attività sospette e impedire audit efficaci.	Log immutabili Protezione con permessi rigorosi e monitoraggio integrità dei log
MU02	URL Malevolo in Messaggi	Spoofing, Information Disclosure	Attaccante esterno che altera comunicazioni o messaggi	Indirizzare utenti verso siti malevoli (phishing) bypassando controlli di sicurezza	Validazione e filtraggio dei link nelle comunicazioni Educazione degli utenti su phishing e sicurezza
MU03	Brute-force su Accesso	Spoofing / Elevation of Privilege	Attaccante esterno che tenta credenziali ripetutamente	Ottenere accessi non autorizzati indovinando username e password	Blocchi temporanei dopo tentativi falliti Multi-factor authentication (MFA, implementazione futura) Monitoraggio
MU04	Denial of Service (DoS)	Denial of Service	Attaccante esterno che genera traffico malevolo massivo	Rendere la piattaforma indisponibile agli utenti legittimi, compromettendo continuità del servizio	Filtri anti-DDoS e rate limiting Bilanciamento del carico e ridondanza infrastrutturale Monitoraggio in tempo reale e risposta automatica (implementazione futura)
MU05	Privilege Escalation	Elevation of Privilege	Utente con permessi limitati che sfrutta vulnerabilità o errori di configurazione	Ottenere privilegi più elevati per manipolare o controllare funzionalità riservate	Controlli rigorosi sui privilegi Monitoraggio e auditing delle attività privilegiate

Tabella 24: Misuse Case rilevanti

L'analisi condotta ha evidenziato una serie di minacce rilevanti per la sicurezza del sistema CyberDefender, che spaziano da attacchi diretti ai meccanismi di autenticazione fino a tentativi di manipolazione dei log, URL malevolo in messaggi e azioni volte a compromettere la disponibilità del servizio.

Tali minacce, se non opportunamente gestite, possono compromettere la riservatezza, integrità e disponibilità del sistema, con impatti potenzialmente critici sia per gli utenti finali che per le infrastrutture aziendali.

Per ciascuna minaccia identificata, sono state proposte misure di mitigazione specifiche, basate su principi di sicurezza proattiva e buone pratiche per rendere il sistema più robusto. L'adozione di tecniche come il monitoraggio continuo, la segregazione dei privilegi e il controllo sull'integrità dei log contribuisce a rafforzare in modo significativo la postura di sicurezza complessiva.

Infine, l'approccio basato sul modello STRIDE ha permesso di classificare e analizzare le minacce in modo sistematico, favorendo una valutazione strutturata dei rischi e supportando la definizione di contromisure efficaci in fase di progettazione e gestione del sistema.

Mitigazioni generali

Nel contesto del progetto CyberDefender, risulta fondamentale l'adozione di un insieme strutturato di contromisure per far fronte alle minacce individuate. Sebbene l'implementazione completa di tutte le mitigazioni descritte sarebbe necessaria per garantire un elevato livello di sicurezza, solo alcune sono state effettivamente realizzate, e in modo semplificato.

Di seguito vengono descritte nel dettaglio le principali misure di mitigazione previste:

1. Autenticazione forte

Prevede l'utilizzo di meccanismi avanzati di verifica dell'identità, per prevenire accessi non autorizzati.

Implementazione attuale: è presente un semplice sistema di login con username e password, in particolare hashata con l'algoritmo Argon2, con blocco con tentativi ripetuti.

2. Controllo dei privilegi

Consiste nell'applicare il principio del minimo privilegio, assegnando a ciascun utente solo i permessi strettamente necessari.

Implementazione attuale: il sistema distingue ruoli base (es. privato/aziendale), ma la gestione fine-grained dei privilegi è assente.

3. Validazione degli input e dei file

Include il controllo di input utente per prevenire l'iniezione di codice e la scansione dei file caricati per rilevare contenuti potenzialmente dannosi.

Implementazione attuale: è presente una validazione basilare lato client, ma non sono ancora attivi antivirus o sandbox per i file.

4. Logging e tracciabilità

I log devono essere sicuri, completi e non modificabili, per garantire la tracciabilità delle azioni e supportare audit e investigazioni.

Implementazione attuale: le azioni principali vengono registrate in log locali, ma non sono protetti da modifiche né centralizzati.

5. Monitoraggio e rilevamento anomalie

Prevede l'uso di strumenti per monitorare costantemente il sistema, rilevare attività sospette e generare alert automatici.

Implementazione attuale: non è ancora presente un sistema di monitoraggio attivo, né una gestione degli eventi centralizzata.

6. Gestione delle vulnerabilità e aggiornamenti

È necessario mantenere il sistema costantemente aggiornato, correggendo vulnerabilità note tramite patch e aggiornamenti software.

Implementazione attuale: gli aggiornamenti vengono effettuati manualmente e in modo non regolare.

La descrizione sopra riportata evidenzia che, pur essendo state definite mitigazioni fondamentali per la protezione del sistema, solo alcune di esse sono state applicate, e in modo parziale o semplificato.

Per garantire una sicurezza robusta e sostenibile nel tempo, sarà necessario estendere e rafforzare le misure attualmente in essere, adottando un approccio più strutturato e completo alla gestione del rischio.

5. Progettazione dell'architettura sicura

CyberDefender è progettata secondo un'architettura **client-server multi-tier**, che separa chiaramente i livelli di interazione con l'utente, logica applicativa e gestione dei dati. Questa divisione consente di garantire scalabilità, sicurezza e facilità di manutenzione, elementi fondamentali per un'applicazione che gestisce dati sensibili e richiede elevati standard di protezione. L'architettura è composta dai seguenti componenti principali:

- **Frontend** (Client Web): Realizzato con Bootstrap (HTML, CSS), il frontend offre un'interfaccia grafica responsiva e intuitiva. Gli utenti possono navigare nei contenuti didattici, riempire form, inserire messaggi o numeri da analizzare e accedere al proprio profilo personale. Tutte le comunicazioni tra frontend e backend avvengono tramite il protocollo http, garantendo riservatezza e integrità dei dati trasmessi. L'autenticazione e l'autorizzazione sono gestite attraverso token sicuri, assicurando sessioni protette.
- **Backend** (Server applicativo): Implementato con Django (Python), il backend gestisce la logica di business, l'elaborazione dei dati, l'analisi dei messaggi sospetti, la generazione dei report personalizzati e la gestione degli utenti. Per l'autenticazione, si sfrutta il sistema integrato di Django, che offre un solido framework per la gestione di utenti, password protette con hashing sicuro, sessioni e protezione CSRF. A questo sistema base, CyberDefender integra funzionalità avanzate come un sistema di controllo accessi basato su ruoli (**RBAC**), per differenziare i permessi tra utenti privati e aziende, garantendo che ciascun utente possa accedere solo alle risorse a lui consentite.
- **Database** (SQLite3): Il database memorizza in modo sicuro tutte le informazioni.
- **Modulo di Analisi Sicura**: Include algoritmi per l'analisi semantica e il riconoscimento di pattern sospetti nei messaggi o nei numeri di telefono forniti dagli utenti.
- **Modulo di Logging e Audit**: Registra in modo sicuro tutte le operazioni rilevanti, quali accessi, modifiche ai dati e analisi effettuate.

Questa architettura è progettata seguendo i principi di **defense-in-depth** e **least privilege**, riducendo la superficie d'attacco grazie all'isolamento delle componenti e all'implementazione di controlli di sicurezza fin dalla fase di progettazione. La modularità della struttura consente di mantenere alta la scalabilità e la facilità di aggiornamento o integrazione di nuove funzionalità in futuro. Infine, i controlli di sicurezza implementati tengono conto delle minacce e mitigazioni identificate nelle analisi preliminari, assicurando che la sicurezza sia parte integrante del sistema e non una componente aggiunta successivamente.

Pattern architetturali adottati

L'architettura di CyberDefender è stata progettata seguendo alcuni pattern architetturali consolidati, con l'obiettivo di garantire modularità, manutenibilità, scalabilità e sicurezza.

1. **Client-Server**: Il sistema adotta il pattern classico Client-Server, dove il client (frontend web) si occupa dell'interazione con l'utente, mentre il server (backend Django) gestisce l'elaborazione, la logica applicativa e l'accesso ai dati. Questo approccio consente una chiara separazione tra presentazione e logica di business.
2. **Three-Tier Architecture**: CyberDefender è strutturata secondo un modello a tre livelli:
 - Presentation Layer: l'interfaccia web responsiva (Bootstrap/HTML).

- Application Layer: il backend Django che gestisce autenticazione, analisi e logica dei permessi.
- Data Layer: il database SQLite3, che memorizza in modo sicuro utenti, log e risultati analitici.

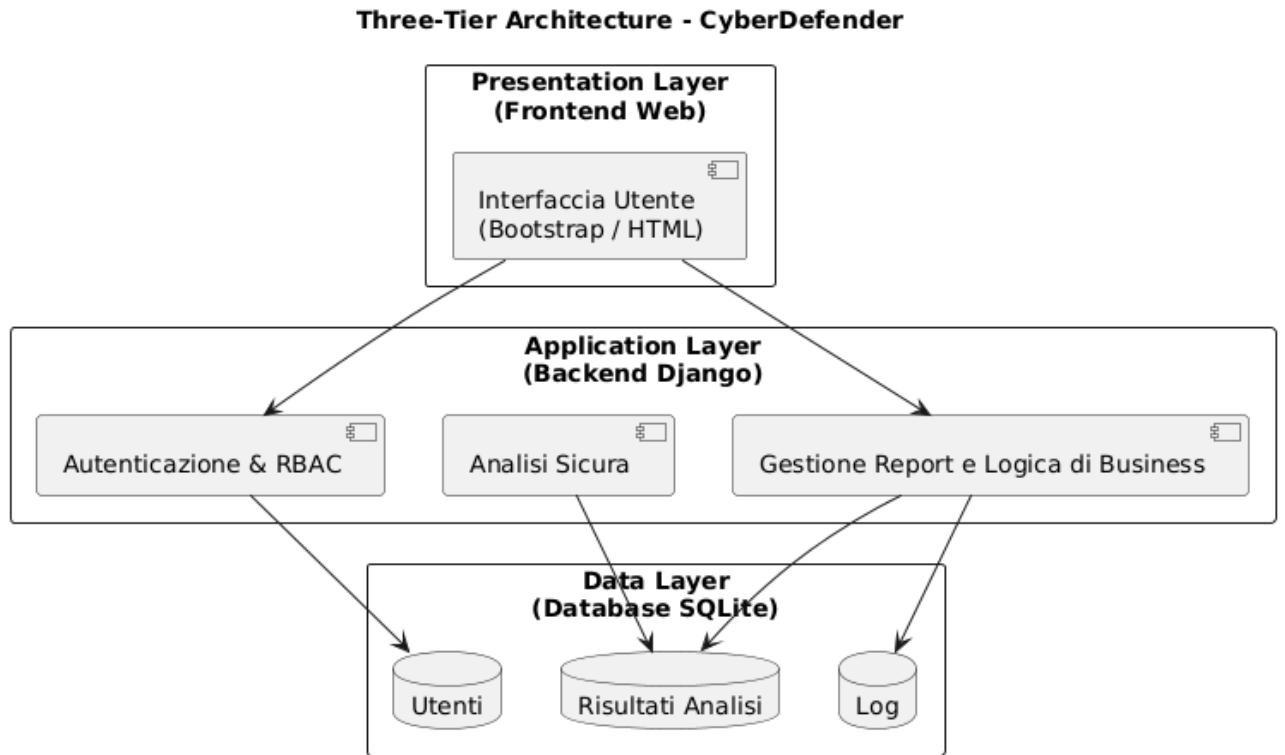


Figura 5: Three-Tier Architecture

Questa suddivisione favorisce la scalabilità, la separazione delle responsabilità e la possibilità di sostituire/modificare ogni livello in modo indipendente.

3. Model-View-Controller (MVC): Il backend Django implementa il pattern MVC, dove:

- Model: rappresenta i dati (utenti, messaggi, risultati).
- View: definisce le risposte HTTP (template).
- Controller (View in Django): gestisce la logica delle richieste utente.

Questo pattern migliora l'organizzazione del codice e separa i dati dalla presentazione.

Security By Design: l'intera architettura segue il principio di Security-by-Design, integrando meccanismi di protezione come:

- Autenticazione con hashing sicuro.
- RBAC (Controllo Accessi Basato su Ruoli).
- Protezione CSRF.
- Logging per audit.

Principi di Progettazione Sicura Applicativa

La progettazione di CyberDefender si basa sui principali principi di progettazione sicura, come indicato nelle linee guida NIST, OWASP e nelle best practice accademiche e industriali. L'applicazione coerente di questi principi consente di ridurre i rischi, mitigare le vulnerabilità e garantire un sistema robusto, resiliente e conforme ai requisiti di sicurezza.

Principio	Applicazione in CyberDefender	Contromisure associate
Least Privilege	Assegnazione di permessi minimi necessari per ogni ruolo e utente.	RBAC (Role-Based Access Control), liste di controllo accessi (ACL)
Defense in Depth	Implementazione di più livelli di sicurezza (frontend, backend, database) per garantire protezione multilivello.	Controlli di accesso, validazione input, protezione CSRF, Argon2.
Fail Securely	Il sistema è progettato per bloccare l'accesso o interrompere operazioni in caso di errori.	Gestione robusta degli errori, logging dettagliato, blocco temporaneo degli account dopo tentativi falliti di login.
Input Validation	Validazione rigorosa di tutti i dati in ingresso, per evitare injection e dati malevoli.	Sanitizzazione input, utilizzo di ORM per query sicure.
Secure Authentication	Uso del sistema di autenticazione Django.	Hashing sicuro delle password (Argon2), gestione sicura delle sessioni, timeout di sessione.
Auditability	Registrazione dettagliata e sicura di tutte le operazioni.	Modulo di logging, audit e monitoraggio.
Session Management	Gestione sicura delle sessioni utente per prevenire hijacking e fixation.	Gestione CSRF, token di sessione.
Error Handling	Comunicazione degli errori senza esporre informazioni sensibili o tecniche.	Messaggi di errore generici per l'utente.

Tabella 25: Principi di progettazione sicura applicativa

L'adozione rigorosa di questi principi garantisce che CyberDefender mantenga elevati standard di sicurezza sin dalla fase di progettazione, minimizzando i rischi di attacchi informatici e proteggendo efficacemente i dati degli utenti.

Vista Architetturale UML

Per rappresentare in modo chiaro e dettagliato la struttura e i flussi interni di CyberDefender, sono stati realizzati diversi diagrammi UML, ognuno focalizzato su un aspetto specifico dell'architettura e del funzionamento del sistema.

- **Diagramma dei Componenti:** Illustra i principali moduli software (frontend, backend, database, moduli di analisi e logging) e le loro relazioni. Il diagramma rappresenta

l'architettura modulare di CyberDefender, organizzata secondo un modello client-server multi-tier.

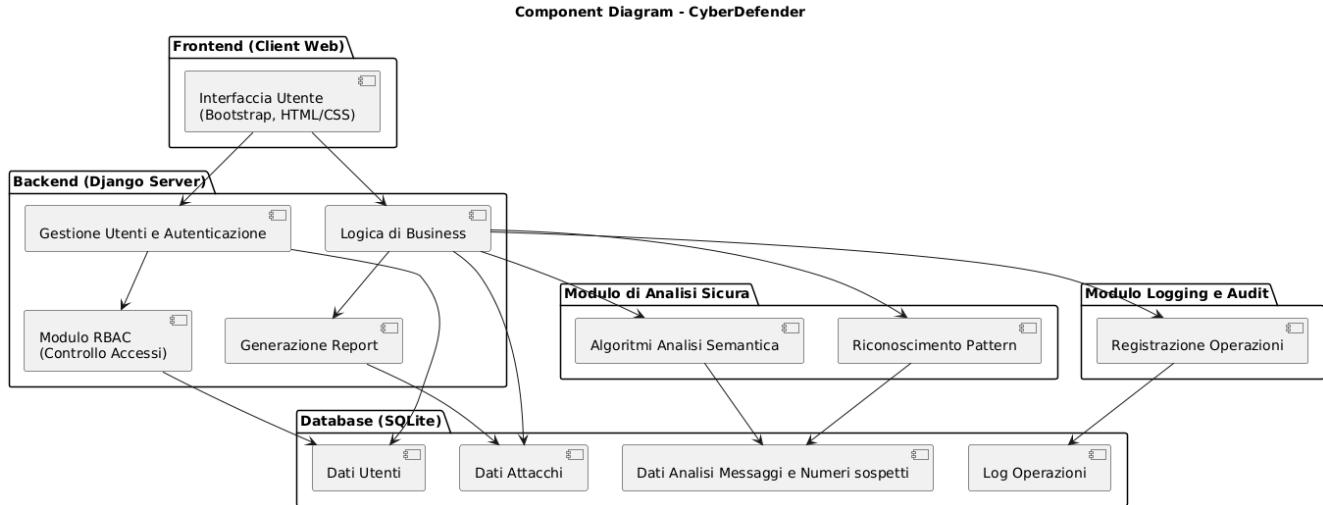


Figura 6: Component Diagram

- **Diagramma di Distribuzione (Deployment Diagram):** Rappresenta la distribuzione fisica dell'applicazione, mostrando i nodi hardware (server, client, database) e come i componenti software sono distribuiti su questi nodi. È utile per analizzare aspetti di sicurezza legati all'infrastruttura.

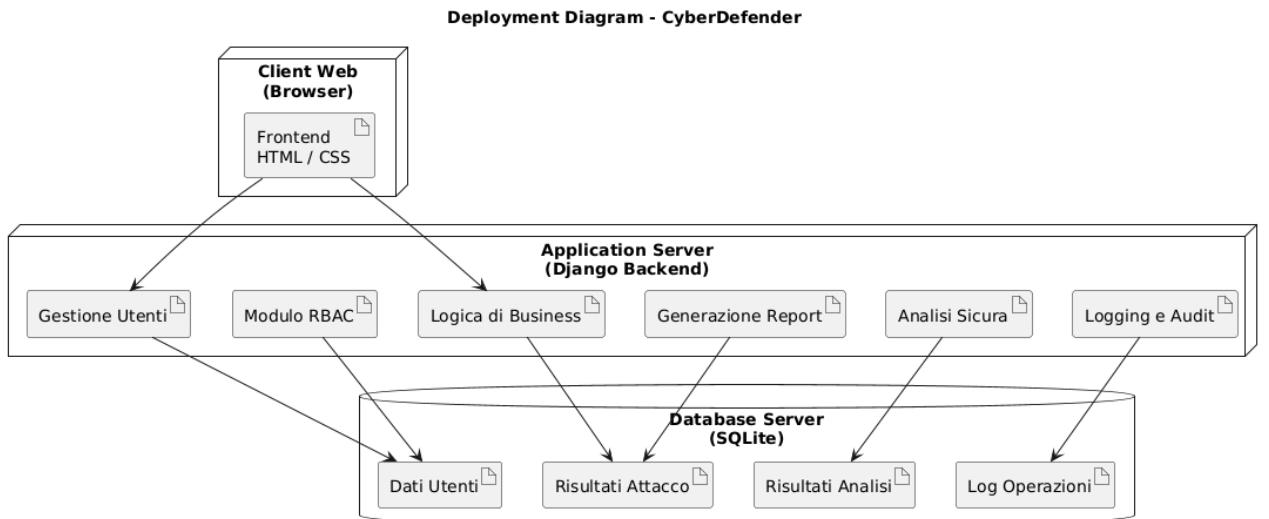


Figura 7: Deployment Diagram

1. **Client Web:** l'utente interagisce tramite un browser, che carica il frontend realizzato in HTML, CSS e JavaScript.
2. **Application Server:** ospita il backend sviluppato in Django.
3. **Database Server (SQLite):** gestisce la persistenza dei dati, tra cui informazioni sugli utenti, log delle operazioni e risultati delle analisi.

- **Diagrammi di Sequenza:** Descrivono i flussi operativi principali, come ad esempio il processo di autenticazione, l'invio di messaggi per l'analisi e la generazione di report. Questi diagrammi evidenziano l'ordine e le interazioni temporali tra oggetti e componenti.

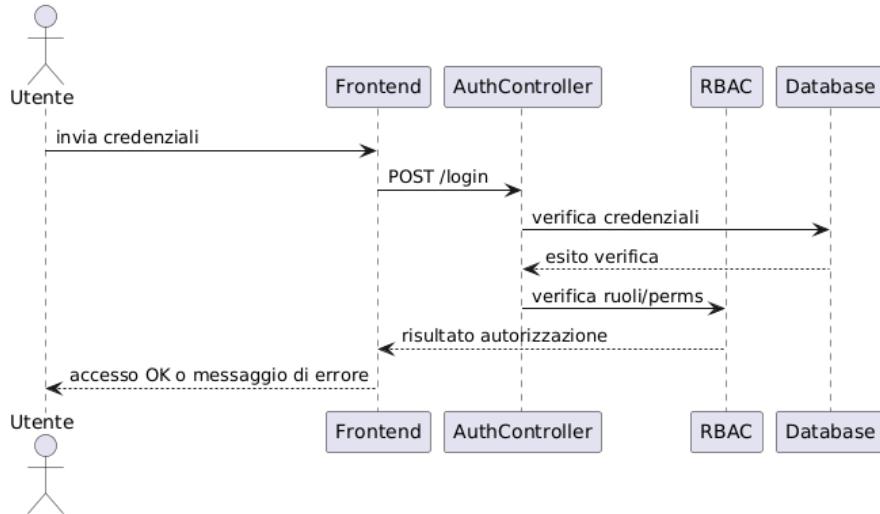


Figura 8: Diagramma di sequenza - Autenticazione (Login)

La figura 8 rappresenta il flusso di autenticazione di un utente. Le fasi principali sono:

1. L'utente inserisce le credenziali di accesso nel frontend.
2. Il frontend invia una richiesta POST al controller di autenticazione (/login).
3. Il controller interroga il database per verificare username e password.
4. Dopo la verifica, il controller interagisce con il modulo RBAC per controllare i ruoli e i permessi dell'utente.
5. Se tutto è corretto, il frontend comunica all'utente l'esito del login, altrimenti viene visualizzato un messaggio di errore.

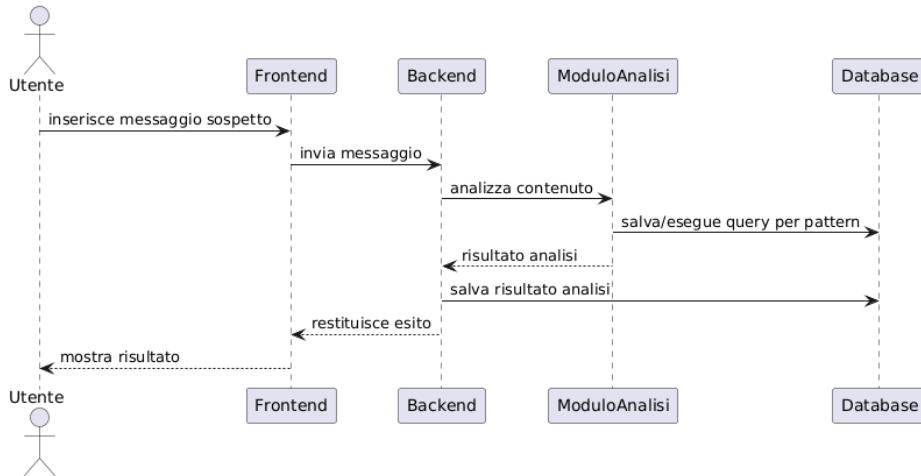


Figura 9: Diagramma di sequenza - Messaggio Sospetto

Questo diagramma rappresenta il flusso in cui un utente invia un messaggio da analizzare. Il frontend inoltra il messaggio al backend, che lo passa al modulo di analisi semantica. Il modulo confronta il contenuto con pattern sospetti e restituisce un esito che viene salvato nel database e poi mostrato all'utente.

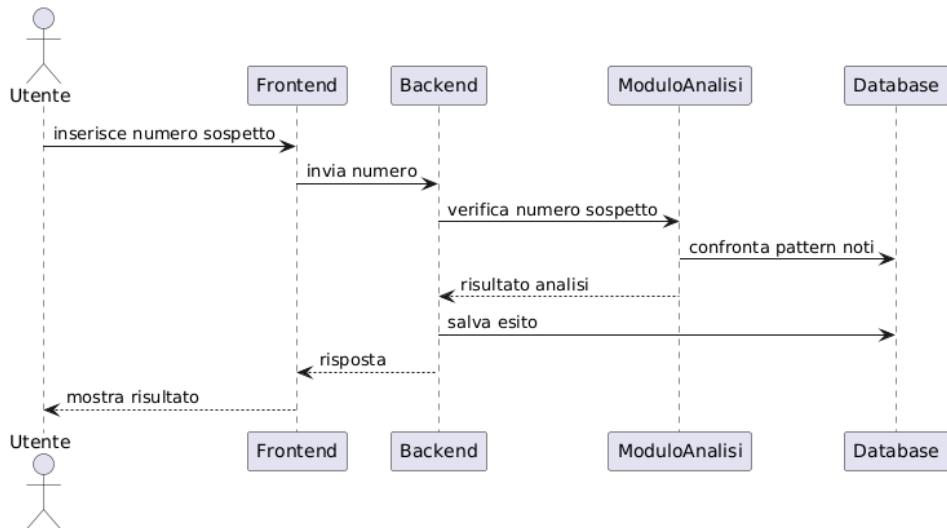


Figura 10: Diagramma di sequenza - Numero Sospetto

L’utente inserisce un numero telefonico che ritiene sospetto. Il backend invia il numero al modulo di analisi, il quale esegue controlli su pattern noti. L’esito dell’analisi viene salvato e restituito all’utente.

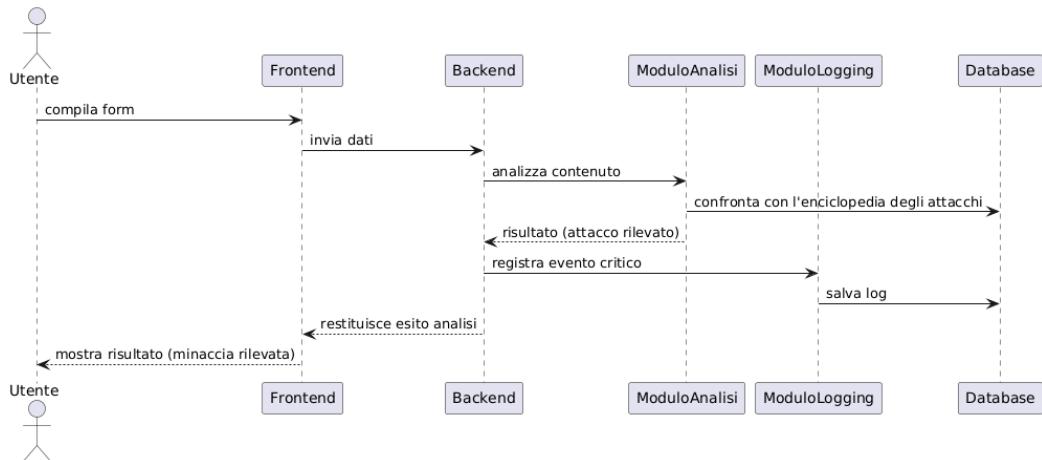


Figura 11: Diagramma di sequenza - Rilevamento Attacco

L’utente compila un form nel frontend. Il frontend invia queste informazioni al backend, che le passa al modulo di analisi. Il modulo confronta i dati con una “enciclopedia degli attacchi” nel database per rilevare eventuali minacce. Se viene identificato un attacco, il backend registra l’evento critico tramite il modulo di logging, salvando il log nel database. Infine, il backend restituisce l’esito dell’analisi al frontend, che mostra il risultato all’utente, segnalando la presenza o meno di una minaccia.

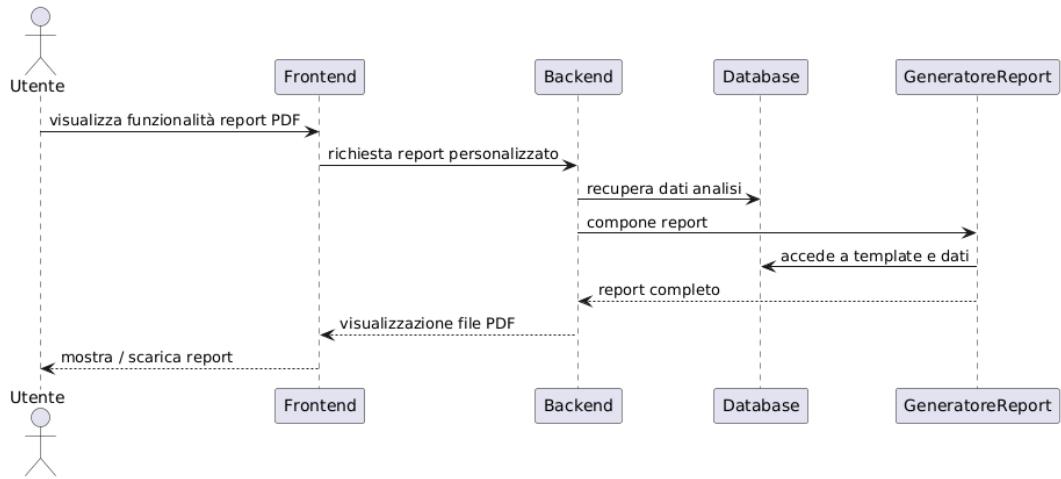


Figura 12: Diagramma di sequenza - Report PDF

L’utente richiede un report personalizzato tramite l’interfaccia. Il backend raccoglie i dati rilevanti dall’analisi del form, e invoca il generatore di report. Il componente crea un aggregando tutte le informazioni necessarie. Il report viene poi restituito al frontend per la visualizzazione o il download da parte dell’utente.

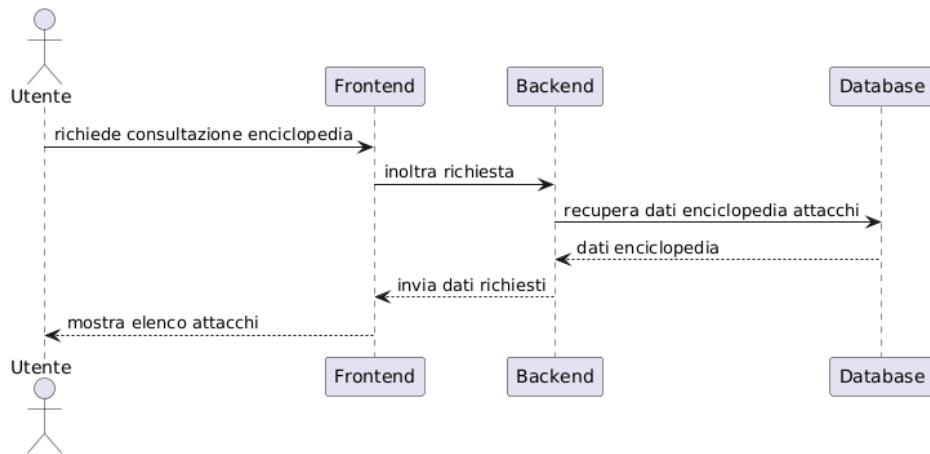


Figura 13: Diagramma di sequenza - Enciclopedia Attacchi

L’utente richiede di consultare l’enciclopedia degli attacchi tramite l’interfaccia frontend. Il frontend invia la richiesta al backend, che recupera dal database le informazioni relative agli attacchi. I dati vengono inviati al frontend, che li visualizza per l’utente.

- **Diagramma delle Attività:** Modella i processi interni e i flussi di lavoro, mettendo in evidenza i punti decisionali e le azioni critiche, importanti per valutare la sicurezza e l’efficienza operativa.

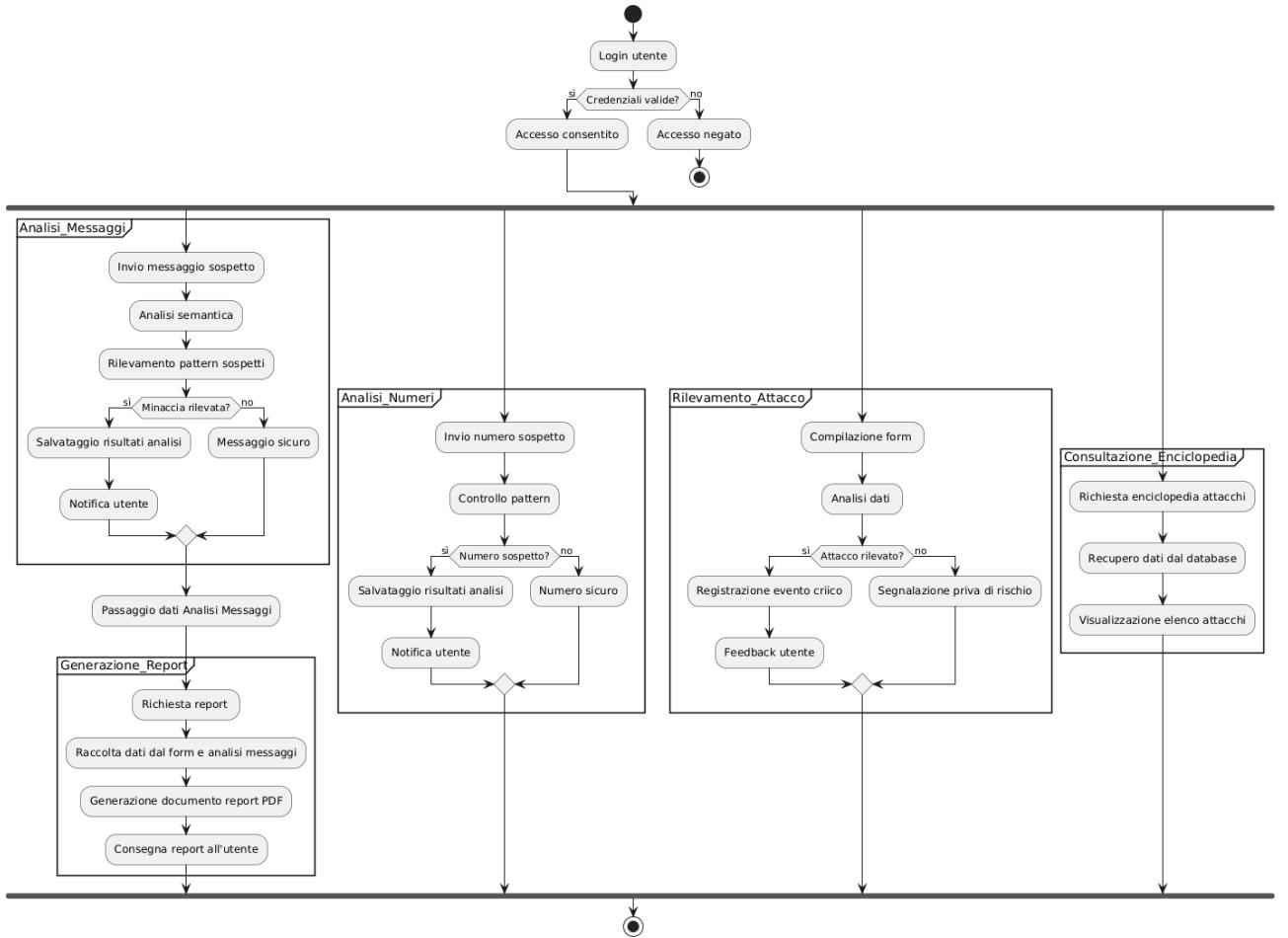


Figura 14: Diagramma Attività

Il diagramma rappresenta il flusso complessivo delle attività di CyberDefender:

1. **Login Utente:** L'utente si autentica. Se le credenziali sono valide, l'accesso è consentito; altrimenti l'accesso è negato e il processo termina.
2. Attività che può svolgere l'utente:
 - **Analisi Messaggi:** L'utente invia messaggi sospetti, che vengono analizzati. Se viene rilevata una minaccia, i risultati vengono salvati e l'utente notificato.
 - **Analisi Numeri:** L'utente invia numeri sospetti per controlli di pattern e riceve notifiche su eventuali rischi.
 - **Rilevamento Attacco:** L'utente compila un form per capire un possibile attacco; se confermato, l'evento critico viene registrato e l'utente riceve feedback.
 - **Generazione Report:** I dati dall'analisi dei form vengono utilizzati per creare e consegnare un report PDF all'utente
 - **Consultazione Enciclopedia:** L'utente può consultare l'elenco degli attacchi memorizzati nel database.

Modello E-R e Logico del Database

Per assicurare una gestione sicura e strutturata dei dati, CyberDefender utilizza un database relazionale SQLite.

Analisi e progettazione concettuale: Modello E/R

Primo prototipo del modello E/R con generalizzazione totale disgiunta. Il progetto prevede l'utilizzo di una generalizzazione totale e disgiunta per l'entità "Utente", al fine di modellare correttamente le specializzazioni logiche previste nel dominio applicativo della web app.

La generalizzazione di *Utente* definisce due sottotipi: Privato e Aziendale.

- **Totale:** ogni istanza dell'entità Utente deve appartenere obbligatoriamente a una delle due specializzazioni.
- **Disgiunta:** un utente può essere o un privato o un utente aziendale, ma non entrambi contemporaneamente.

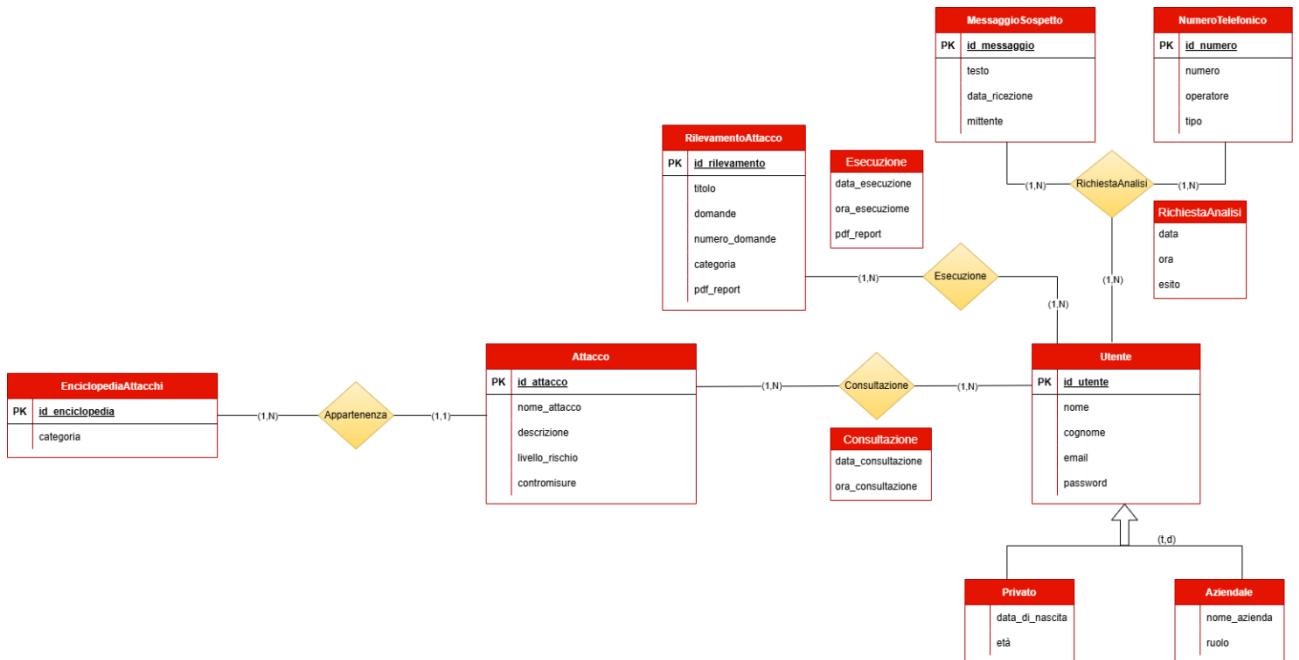


Figura 15: Modello E/R

Possibili soluzioni della generalizzazione

- 1) **Strategia "Tutto nel padre":** La generalizzazione ingloba completamente le specializzazioni, assorbendo gli attributi delle entità figlie. È una soluzione conveniente quando le operazioni non differenziano tra le occorrenze delle diverse entità figlie. Tuttavia, può introdurre valori NULL per attributi non applicabili a tutti i tipi.
- Utente: (ID_utente (PK), Nome, Cognome, Email, Password, TipoUtente (enum: "Aziendale", "Privato", NomeAzienza, Ruolo, DataDiNascita, Età))
 - Vantaggi:** Meno tabelle, modello più semplice
 - Svantaggi:** Presenza di molti valori NULL se alcuni attributi non si applicano a tutte le entità.

2) Strategia "Tutto nelle figlie": Consiste nell'accentramento dei dati direttamente nelle entità figlie. Le specializzazioni ereditano gli attributi della generalizzazione, che vengono replicati in ciascuna entità figlia. È una strategia applicabile solo in caso di generalizzazione totale (cioè ogni istanza dell'entità padre appartiene necessariamente a una figlia). È utile quando le operazioni da svolgere sono diverse per ciascun tipo specializzato.

- Aziendale: (NomeAzienda, Ruolo, ID_utente (PK), Nome, Cognome, Email, Password)
- Privato: (DataDiNascita, Età, ID_utente (PK), Nome, Cognome, Email, Password)

Vantaggi: Nessun valore NULL, poiché ogni attributo è pertinente all'entità.

Svantaggi: Maggior numero di tabelle

Modello E/R ottimizzato

Nel passaggio al modello relazionale, è stato scelto di unificare le entità specializzate con la super-entità. Questo comporta una sola tabella per “Utente”, con attributi opzionali specifici per Privato e Aziendale, con l'aggiunta dell'attributo **TipoUtente** per distinguerli. Questa scelta è coerente con una generalizzazione totale (nessun valore orfano) e disgiunta (nessuna sovrapposizione), e consente una gestione più semplice delle entità nel database.

Questa decisione progettuale è stata motivata dal fatto che le funzionalità previste dal sistema non richiedono la gestione separata delle diverse tipologie di utente. Gli attributi delle entità figlie della tabella “Utente” (Privato, Aziendale) vengono inglobati nell'entità padre “Utente”. Per gli utenti che non rientrano in una delle due specializzazioni, gli attributi non pertinenti assumono valore “NULL” nel database relazionale risultante.

- Utente: (ID_utente (PK), Nome, Cognome, Email, Password)
 - Aziendale: (NomeAzienda, Ruolo)
 - Privato: (DataDiNascita, Età)

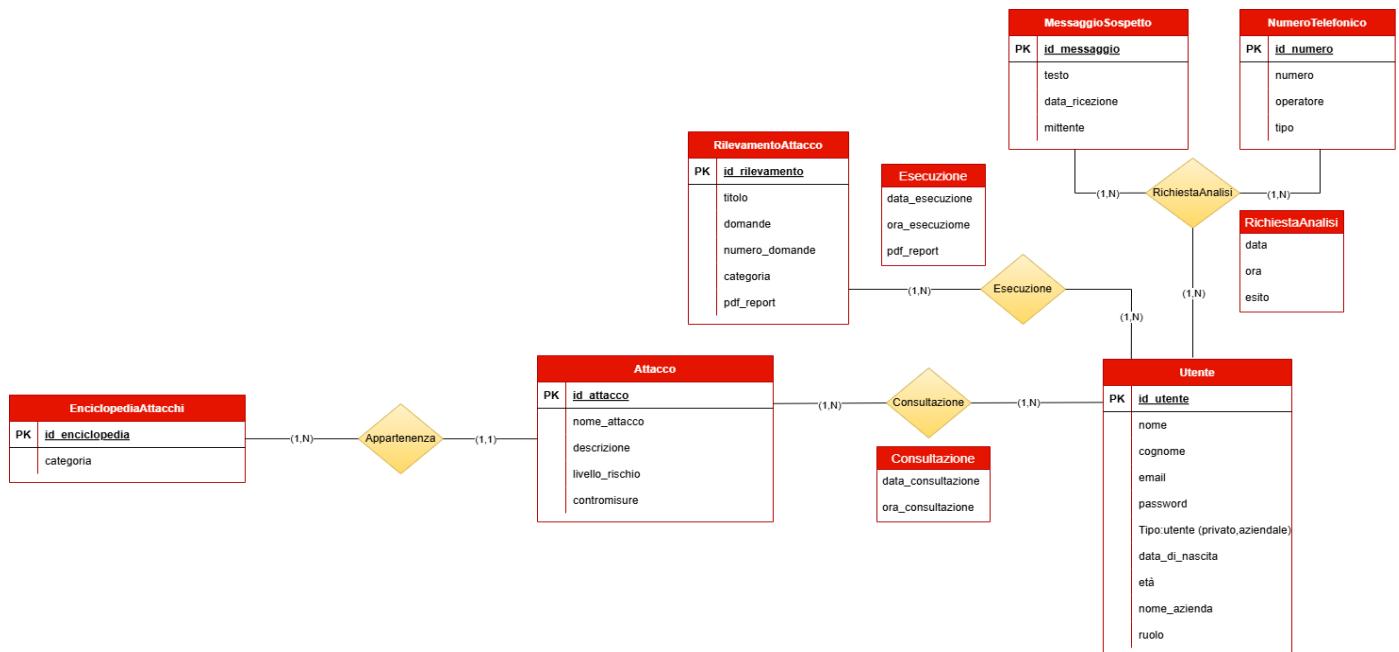


Figura 16: Modello E/R ottimizzato

Nel modello E/R questi attributi saranno visualizzati come facenti parte dell'entità Utente, ma vincolati dalla specializzazione.

Entità Principali

Utente (generalizzazione): Rappresenta chi utilizza la web app. Può essere un utente aziendale o privato.

Attributi: ID_utente (PK), Nome, Cognome, Email, Password

Specializzazioni:

- Aziendale: NomeAzienda, Ruolo
- Privato: DataDiNascita, Età

EnciclopediaAttacchi: Archivio contenente informazioni sugli attacchi malevoli, una descrizione, esempi e contromisure.

Attributi: ID_enciclopedia (PK), Categoria

Attacco: Attacco consultato dall'utente

Attributi: ID_attacco (PK), NomeAttacco, Descrizione, LivelloRischio (enum: "Basso", "Medio", "Alto"), Contromisure

RilevamentoAttacco: Contenuto interattivo volto a riconoscere il tipo di attacco ricevuto.

Attributi: ID_rilevamento (PK), Titolo, Domande, NumeroDomande, Categoria, PDF_report.

MessaggioSospetto: Messaggi analizzati e considerati sospetti o malevoli dati come input da analizzare al sistema.

Attributi: ID_messaggio (PK), Testo, DataRicezione, Mittente

NumeroTelefonico: Numeri telefonici analizzati nel sistema.

Attributi: ID_numero (PK), Numero, Operatore, Tipo (enum: "mobile", "fisso")

Relazioni

Consultazione: L'azione o processo con cui un utente consulta l'enciclopedia degli attacchi.

Attributi: DataConsultazione, OraConsultazione

Appartenenza: Rappresenta l'appartenenza dell'attacco all'enciclopedia degli attacchi.

RichiestaAnalisi: Processo o azione di analisi su numero telefonico o messaggio malevolo.

Attributi: Data, Ora, Esito (es. "Sicuro", "Malevolo")

Esecuzione: Rappresenta l'azione con cui un utente svolge un rilevamento di un attacco.

Attributi: Data_Esecuzione, OraEsecuzione, PDF_report

Progettazione Logica

La progettazione logica della web app è stata realizzata sulla base del modello concettuale E/R precedentemente descritto, adottando una generalizzazione totale e disgiunta per l'entità "Utente" (specializzata in Privato e Aziendale).

Utente: (ID_utente (PK), Nome, Cognome, Email, Password, TipoUtente (enum: "Aziendale", "Privato"), NomeAzienda, Ruolo, DataDiNascita, Età)

- Per incorporare le specializzazioni nel padre è stato aggiunto l'attributo *TipoUtente*.

Rilevamento attacco: (ID_rilevamento (PK), Titolo, Domande, NumeroDomande, Categoria, PDF_report)

Esecuzione: (ID_utente : Utente (UNIQUE), (ID_rilevamento: RilevamentoAttacco (UNIQUE), Data_Esecuzione, OraEsecuzione, PDF_report)

MessaggioSospetto: (ID_messaggio (PK), Testo, DataRicezione, Mittente)

NumeroTelefonico: (ID_numero (PK), Numero, Operatore, Tipo (enum: "mobile", "fisso"))

RichiestaAnalisi: (ID_messaggio : MessaggioSospetto (UNIQUE), ID_numero : NumeroTelefonico (UNIQUE), Data, Ora, Esito)

EnciclopediaAttacchi: (ID_encyclopedia (PK), Categoria)

Attacco: (ID_attacco (PK), NomeAttacco, Descrizione, LivelloRischio (enum: "Basso", "Medio", "Alto"), Contromisure, ID_encyclopedia : Enciclopedia)

- Poiché tra "EnciclopediaAttacchi" e "Attacco" c'è una relazione (N,1) inserisco la chiave primaria di EnciclopediaAttacco in Attacco.

Consultazione: (ID_attacco : Attacco, ID_utente : Utente, DataConsultazione, OraConsultazione)

6. Sviluppo sicuro del software

Per implementare le funzionalità software, il team ha seguito un approccio Secure by Design.

Scelte tecnologiche e linguaggio

Componente	Linguaggio/Framework	Versione	Note
Backend	Django	v5.2.4	Framework semplice da utilizzare ma con un ORM molto potente, promuove uno sviluppo rapido con un codice pulito e riutilizzabile.
Frontend	Bootstrap CSS, HTML, Django Template Model	Bootstrap v5.3	Bootstrap consente una veloce implementazione di elementi con classi predefinite
Database	SQLite	v3.49.1	Database predefinito di Django, garantisce un buon livello di sicurezza anche senza configurazioni specifiche
Sicurezza	argon2-cffi	v25.1.0	Libreria che implementa l'algoritmo vincitore del Password Hashing Competition nel 2015 rendendolo semplice da applicare
Sicurezza	Django Object-Relational Mapper	Integrato	Costruisce in automatico delle query SQL sicure da SQLi
Sicurezza	Django's Form Class	Integrato	Django fornisce una classe che valida e pulisce l'input utente prima di usarlo.
Sicurezza	Sistema di autenticazione Django	Integrato	Fornisce sia autenticazione che autorizzazioni. Fornisce anche decorators per proteggere le viste e verificare i permessi.

Tabella 26: Scelte tecnologiche e di linguaggio

Best Practices di programmazione sicura

Pratica	Applicazione in CyberDefender
Input Validation	Vengono usate le classi specializzate di Django (Form class) e alcune estensioni personalizzate di queste ultime, ove necessario.
Uso di librerie sicure e aggiornate	Vengono usate solo librerie python tra le più famose e utilizzate da tutti, aggiornandole costantemente all'ultima versione con script specifici
Separazione logica dei moduli	Ogni funzionalità è stata inserita in un modulo diverso (in django rappresentato dalle app) per gestire e manutenere il codice in maniera separata
Token anti-CSRF e XSS	Django consente di integrare in automatico valori come token anti-CSRF in ogni form, aiutando lo sviluppatore nel processo
Log di sicurezza	Sono stati configurati diversi file di log per tenere traccia di ogni evento, in base alla gravità.
Errori Generici	Ogni eccezione lanciata dal sistema verrà gestita singolarmente e il sistema mostrerà sempre errori generici all'utente
Uso ambienti virtuali	Sono stati configurati diversi ambienti virtuali separati tra loro, uno per ogni ambiente di sviluppo

Tabella 27: Programmazione Sicura

Gestione degli errori ed eccezioni

Gli errori e le eccezioni sono state gestite in modo da assicurare un'esperienza sicura e ottimale in CyberDefender.

Principi Adottati:

- **Prevenzione dei crash:**

Ogni funzione intercetta quante più eccezioni possibili e gestisce anche gli errori imprevisti con funzionalità di default, evitando l'interruzione del sistema.

- **Log dettagliati:**

Ogni errore viene registrato secondo vari criteri in log separati per ogni funzionalità. I criteri includono la gravità dell'evento, il timestamp, il processo o thread che l'ha generato e soprattutto il modulo da cui proviene

- **Error feedback all'utente:**

Ogni messaggio di errore viene presentato all'utente secondo le viste dell'interfaccia grafica, invitando l'utente ad utilizzare l'applicazione in modo corretto.

Esempio gestione errori in homepage

```
def render_homepage(request):
    user_id = request.session.get('user_session_id')
    if user_id:
        try:
            user_data = Utente.objects.get(id=user_id)
        except Utente.DoesNotExist:
            request.session.flush()
            return redirect('')
    reports = Esecuzione.objects.filter(utente=user_data) \
        .select_related('rilevamento_attacco') \
        .order_by('-data_esecuzione', '-ora_esecuzione')

    consultazioni = ConsultazioneAttacco.objects.filter(utente=user_data) \
        .select_related('attacco') \
        .order_by('-data_consultazione', '-ora_consultazione')

    richieste = RichiestaAnalisi.objects.select_related('messaggio_sospetto', 'numero_telefonico') \
        .order_by('-data_richiesta', '-ora_richiesta')

    return render(request, 'homepage.html', {
        'data': user_data,
        'reports': reports,
        'consultazioni': consultazioni,
        'richieste': richieste
    })
else:
    return render(request, 'loginIndex.html', {'error_message': "Esegui il login prima di accedere alla homepage"})
```

La funzione `render_homepage` gestisce il rendering della pagina di homepage, ma solo ed esclusivamente se l'utente ha una sessione attiva e valida. In caso contrario, esegue il redirect alla pagina di login e mostra un messaggio di errore.

Esempio gestione errori in registrazione utente

```
def registrazione_privato(request):
    if request.method == 'POST':
        form = privateRegistrationForm(request.POST)
        email = request.POST.get('email')
        utente = Utente.objects.filter(email=email)
        if len(utente) != 0:
            return render(request, 'registra_privato.html', {'error_message': "Questa mail è già associata ad un altro utente."})

        if form.is_valid():
            hashed_password = make_password(form.cleaned_data['password'])

            utente = Utente.objects.create(
                email=form.cleaned_data['email'],
                password=hashed_password,
                data_nascita=form.cleaned_data['data_nascita'],
                nome=form.cleaned_data['nome'],
                cognome=form.cleaned_data['cognome'],
                tipo_utente='privato',
                ruolo='privato',
            )
            try:
                utente.full_clean()
            except ValidationError:
                utente.delete()
                return render(request, 'loginIndex.html', {'error_message': "Dati inseriti non validi"})

            return render(request, 'loginIndex.html', {'success_message': "Registrazione avvenuta con successo"})

    else:
        form = privateRegistrationForm()
    return render(request, 'registra_privato.html', {'form': form})
```

In questo esempio, la funzione `registrazione_privato` esegue un controllo sia sulla validità dei campi del form, sia sulla buona riuscita dell'operazione di creazione utente. Se l'operazione va male, si esegue un rollback forzato sulla tabella utente e viene mostrato un errore di all'utente.

Esempio logging durante la fase di login utente

```
try:  
    user_data = Utente.objects.filter(email=email).get()  
except Utente.DoesNotExist:  
    logger.info( str(datetime.now()) + " login errato: Email Errata (" + email + ")" )  
    return render(request, 'loginIndex.html', {'error_message' : "Email e/o password non validi"})
```

In questa parte della funzione **checkLogin**, che si occupa di far accedere l'utente al sistema, viene registrato nel log ogni evento di inserimento mail errata. Il risultato nel file login.log è:

```
[INFO] 2025-07-24 02:10:44.426430 login errato: Email Errata (aaaaaaaaaaaaaaaaaaaaaaa@gmail.com)  
[INFO] 2025-07-24 11:18:03.333574 login errato: Email Errata (dada@gmail.com)
```

Errori in GUI durante la registrazione

Nome:

Cognome:

Email:

• Enter a valid date.

Data nascita: 

• La password deve contenere almeno 8 caratteri.

Password:

Registrati

• Enter a valid date.
• La password deve contenere almeno 8 caratteri.

L'esempio mostra il messaggio che l'utente visualizza se non inserisce dati corretti durante la registrazione.

Autenticazioni e autorizzazioni

In CyberDefender le funzionalità possono essere utilizzate solo da utenti registrati, questo per garantire integrità all'interno dei dati del database.

Esempio controllo della sessione utente

```
def numeri_index(request):  
    if request.session.get('user_session_id'):br/>        return render(request, 'analisi_numeri.html')  
    else:  
        return render(request, 'analisi_numeri.html', {'message': "Per eseguire questa funzionalità è necessario eseguire il login."})
```

La funzione si occupa di visualizzare la pagina HTML dell'analizzatore di numeri telefonici, ma la carica completamente solo se l'utente ha già eseguito il login, in caso contrario visualizzerà

un messaggio di errore all'utente. Questo viene implementato tramite la funzione get() fornita da django.contrib.sessions, che restituisce null se la sessione richiesta non viene trovata.

Rate Limiting e durata massima sessione

```
def checkLogin(request):
    request.session.set_expiry(300)

    if request.method == "POST":
        failed_attempts = request.session.get('failed_login_attempts', 0)

        if failed_attempts >= 3:
            return render(request, 'loginIndex.html', {
                'error_message': "Hai superato il numero massimo di tentativi. Riprova più tardi."
            })

        email = request.POST.get("email")
        password = request.POST.get('password')

        try:
            user_data = Utente.objects.filter(email=email).get()
        except Utente.DoesNotExist:
            request.session['failed_login_attempts'] = failed_attempts + 1
            logger.info(str(timezone.now()) + " login errato: Email Errata (" + email + ")")
            return render(request, 'loginIndex.html', {'error_message': "Email e/o password non validi"})

        hashed_password = user_data.password

        if check_password(password, hashed_password):
            request.session['failed_login_attempts'] = 0
            request.session['user_session_id'] = user_data.id

            reports = Esecuzione.objects.filter(utente=user_data) \
                .select_related('rilevamento_attacco') \
                .order_by('-data_esecuzione', '-ora_esecuzione')

            consultazioni = ConsultazioneAttacco.objects.filter(utente=user_data) \
                .select_related('attacco') \
                .order_by('-data_consultazione', '-ora_consultazione')

            richieste = RichiestaAnalisi.objects.select_related('messaggio_sospetto', 'numero_telefonico') \
                .order_by('-data_richiesta', '-ora_richiesta')

            return render(request, 'homepage.html', {
                'data': user_data,
                'reports': reports,
                'consultazioni': consultazioni,
                'richieste': richieste
            })
        else:
            request.session['failed_login_attempts'] = failed_attempts + 1
            return render(request, 'loginIndex.html', {'error_message': "Email e/o password non validi"})
```

La funzione imposta un tempo massimo prima che la sessione venga eliminata automaticamente (300 secondi = 5 minuti) e imposta un nuovo session cookie “**failed_login_attempts**” e incrementa questo valore ogni volta che l’utente sbaglia email o password. Dopo che la variabile arriva a 3, l’utente viene bloccato dall’eseguire il login. Questa tecnica è necessaria per contrastare attacchi a forza bruta alla funzione di login.

Gestione sicura della cifratura delle password

```
hashed_password = make_password(form.cleaned_data['password'])
```

- **Cifratura con Argon2**

Ogni password viene hashata secondo l'algoritmo Argon2id, che garantisce un ottimo rapporto sicurezza/performance essendo progettato per resistere sia ad attacchi di cracking GPU che attacchi side-channel.

- **Funzione predefinita Django**

La funzione che si occupa di gestire l'hashing è una funzione fornita da Django: “**make_password()**”. La funzione fa parte della classe “**django.contrib.auth.hashers**”, da cui deriva anche la funzione per testare l'hash di una password con una password in chiaro: “**check_password()**”.

Esempio di utilizzo check_password nel login

```
hashed_password = user_data.password

if check_password(password, hashed_password):
    request.session['failed_login_attempts'] = 0
    request.session['user_session_id'] = user_data.id

    reports = Esecuzione.objects.filter(utente=user_data) \
        .select_related('rilevamento_attacco') \
        .order_by('-data_esecuzione', '-ora_esecuzione')

    consultazioni = ConsultazioneAttacco.objects.filter(utente=user_data) \
        .select_related('attacco') \
        .order_by('-data_consultazione', '-ora_consultazione')

    richieste = RichiestaAnalisi.objects.select_related('messaggio_sospetto', 'numero_telefonico') \
        .order_by('-data_richiesta', '-ora_richiesta')

    return render(request, 'homepage.html', {
        'data': user_data,
        'reports': reports,
        'consultazioni': consultazioni,
        'richieste': richieste
    })
else:
    request.session['failed_login_attempts'] = failed_attempts + 1
    return render(request, 'loginIndex.html', {'error_message': "Email e/o password non validi"})
```

La funzione **check_password** controlla se l'hash della password inserita dall'utente nel form corrisponde all'hash della password presente nel database: se corrisponde allora viene eseguita tutta la logica per visualizzare la homepage, altrimenti l'utente riceve un errore di login non specifico, così da restituire meno informazioni possibili.

Verifica di sicurezza automatica

Durante lo sviluppo di CyberDefender si è scelto di utilizzare strumenti automatici per il controllo delle dipendenze, allo scopo di rilevare eventuali punti critici del sistema e sviluppare una patch per mantenere alto il livello di sicurezza. Uno degli strumenti utilizzati è stato **bandit**. Il risultato completo dello strumento è stato:

```
Code scanned:  
    Total lines of code: 657207  
    Total lines skipped (#nosec): 2  
  
Run metrics:  
    Total issues (by severity):  
        Undefined: 0  
        Low: 2832  
        Medium: 249  
        High: 45  
    Total issues (by confidence):  
        Undefined: 0  
        Low: 14  
        Medium: 85  
        High: 3027  
Files skipped (0):
```

Lo screen mostra che sono state scannerizzate 657.207 righe di codice, e sono state trovati diversi problemi critici.

Questo strumento ha permesso al team di rivedere le funzionalità implementate secondo un nuovo punto di vista e correggere gli errori per migliorare la sicurezza complessiva di CyberDefender.

7. Testing e verifica del software

Il processo di testing di **CyberDefender** è stato progettato per garantire l'affidabilità, la qualità del codice e la conformità ai requisiti funzionali e di sicurezza. Sono stati adottati approcci sistematici e metodologie avanzate per assicurare che l'applicazione sia robusta, sicura e performante, sia per gli utenti privati che aziendali. I test sono stati condotti su più livelli, coprendo tutte le funzionalità principali della piattaforma.

Obiettivi del testing

Gli obiettivi principali del processo di testing di **CyberDefender** sono finalizzati a garantire che l'applicazione soddisfi pienamente i requisiti funzionali, non funzionali e di sicurezza. In particolare, il testing ha come scopo:

1. **Verifica della correttezza funzionale:** Assicurarsi che tutte le funzionalità offerte da CyberDefender (come l'analizzatore di messaggi sospetti, la verifica dei numeri di telefono, la compilazione del form con la successiva generazione di report PDF e la consultazione dell'enciclopedia) si comportino come previsto, secondo i requisiti raccolti.
2. **Validazione dell'esperienza utente (UX):** Garantire che l'interfaccia utente sia chiara, accessibile e coerente con le esigenze di utenti privati e aziendali, offrendo un'esperienza fluida e intuitiva.
3. **Affidabilità e stabilità dell'applicazione:** Rilevare e correggere eventuali bug, crash o malfunzionamenti che potrebbero compromettere il corretto utilizzo della piattaforma nel tempo.
4. **Sicurezza del sistema:** Verificare la protezione dell'app contro le principali vulnerabilità note (come SQL injection, XSS, CSRF), assicurando la riservatezza, l'integrità e la disponibilità dei dati, in conformità alle linee guida OWASP.
5. **Prestazioni e scalabilità:** Misurare i tempi di risposta delle funzionalità principali e testare il comportamento dell'app sotto carichi elevati, per garantirne l'efficienza e l'adattabilità anche in scenari ad alta richiesta.
6. **Conformità ai requisiti:** Garantire che ogni requisito definito nella fase di analisi (funzionale e non funzionale) sia stato implementato correttamente e testato, assicurando una completa copertura del sistema.
7. **Supporto al miglioramento continuo:** Fornire feedback misurabili e concreti al team di sviluppo per migliorare iterativamente la qualità del codice, l'interfaccia e l'architettura complessiva del progetto.

Metodologie Adottate

Sono stati impiegati i seguenti tipi di test:

- **Unit Test:** Ogni componente del backend (Django) è stato testato isolatamente per verificarne il comportamento atteso. Le unità testate includono modelli, viste, funzioni di utilità e API.

- **Integration Test:** Verificano l'interazione tra i vari moduli del sistema, come la comunicazione tra backend e frontend, tra moduli di analisi e generazione report, e tra utente e sistema attraverso i form di input.
- **System Test:** L'intera applicazione è stata testata come un sistema completo, simulando scenari d'uso reali, per validare il comportamento complessivo della piattaforma rispetto ai requisiti funzionali.
- **Acceptance Test:** Sono stati sviluppati scenari di test sulla base dei requisiti dell'utente (sia privato che aziendale). Questi scenari sono stati utilizzati da utenti reali o simulati per validare l'esperienza utente, le funzionalità interattive e la facilità d'uso.
- **Penetration Test:** Sono stati eseguiti test per valutare la resistenza dell'app a tentativi di attacco, inclusi SQL Injection, Cross-Site Scripting (XSS), Cross-Site Request Forgery (CSRF), brute-force e session hijacking.

Tipologie di test

Questa sezione descrive le principali tipologie di test effettuate per verificare l'implementazione corretta e sicura di tutti i requisiti funzionali (RF), non funzionali (RNF), user stories (US) e misuse case (MC). Per ciascun requisito/test è stato definito un metodo di verifica e monitorato lo stato di completamento.

Test Suite	Metodi di Verifica	Stato
TS-01	login/tests.py	Superato
TS-02	analisiMessaggi/tests.py	Superato
TS-03	enciclopedia/tests.py	Superato
TS-04	enciclopedia/tests.py	Superato
TS-05	analisi_numeri/tests.py	Superato
TS-06	enciclopedia/tests.py	Superato

Tabella 28: Tipologia test

Test Misuse Case

ID	Misuse Case	Strategia di Test	Obiettivo	Risultato
MU01	Manipolazione Log di Sistema	Tentativi di accesso a funzionalità o dati riservati a ruoli superiori	Modificare o cancellare log per nascondere attività sospette e impedire audit efficaci.	Il sistema deve impedire l'accesso e registrare l'evento nei log di sicurezza

ID	Misuse Case	Strategia di Test	Obiettivo	Risultato
MU02	URL Malevolo in Messaggi	Attaccante esterno che inserisce un input non conforme ai requisiti	Verifica che l'accesso sia impedito	Il sistema deve impedire l'accesso a determinate pagine e reindirizzare l'utente alla pagina di login
MU03	Brute-force su Accesso	Attaccante esterno che tenta credenziali ripetutamente	Ottener accessi non autorizzati indovinando username e password	Blocchi temporaneo dell'accesso dopo un numero di tentativi definiti
MU05	Privilege Escalation	Utente con permessi limitati che sfrutta vulnerabilità o errori di configurazione	Ottener privilegi più elevati per manipolare o controllare funzionalità riservate	L'utente che non ha i permessi adeguati non può accedere alle funzionalità del sistema

Tabella 29: Test Misuse Case

Organizzazione dei test: Test Suite e Test Case

I test sviluppati per il sistema CyberDefender sono stati organizzati in Test Suite, ognuna delle quali raccoglie un insieme di Test Case relativi a uno o più requisiti presenti nella Tabella precedenti. Ogni test case è stato progettato per validare in modo preciso un comportamento atteso del sistema, seguendo una struttura standardizzata che include:

- **ID:** identificatore univoco del test case
- **Nome:** breve descrizione della funzionalità verificata
- **Input:** dati o condizioni iniziali richieste per eseguire il test
- **Output Atteso:** comportamento atteso del sistema
- **Stato:** esito dell'esecuzione

Esempio di Test Suite: AUTENTICAZIONE UTENTE (TS01)

ID	Nome Test Case	Input	Output Atteso	Stato
TC01	Login con credenziali valide	Email registrata, password corretta	Accesso consentito e redirect alla dashboard	Superato

ID	Nome Test Case	Input	Output Atteso	Stato
TC02	Login con password errata	Email registrata, password errata	Messaggio di errore e blocco accesso	Superato
TC03	Login con utente non esistente	Email non registrata	Messaggio "utente non trovato"	Superato
TC04	Logout	Sessione attiva	Chiusura sessione e redirect alla home	Superato
TC05	Tentativi ripetuti login	>3 tentativi falliti consecutivi	Blocco temporaneo IP / Captcha attivato	Superato
TC06	Dati sensibili protetti da hashing (Argon2)	Password salvata nel database hashata	Impossibile visualizzare le password in chiaro	Superato

Tabella 30: TS01

Esempio di Test Suite: ANALISI MESSAGGI SOSPETTI (TS02)

ID	Nome Test Case	Input	Output Atteso	Stato
TC07	SQL Injection test	Input ' OR '1'='1 in form login	Input bloccato e log attività sospette	Superato

Tabella 31: TS02

Esempio di Test Suite: ENCICLOPEDIA DEGLI ATTACCHI (TS03)

ID	Nome Test Case	Input	Output Atteso	Stato
TC08	Visualizzazione scheda	Selezione della scheda da visualizzare	Visualizzazione della scheda	Superato
TC09	Verifica completezza contenuti schede	Accesso a più schede	Ogni scheda mostra tutte le sezioni (definizione, modalità, conseguenze, consigli)	Superato
TC10	Verifica linguaggio accessibile	Accesso da utente non tecnico	Linguaggio semplice, termini tecnici spiegati	Superato
TC11	UI chiara, accessibile e reattiva	Navigazione Utente	Interfaccia semplice con cui interagire	Superato

Tabella 32: TS03

Esempio di Test Suite: REPORT PDF CONSIGLI SU MISURA (TS04)

ID	Nome Test Case	Input	Output Atteso	Stato
TC12	Generazione PDF dopo analisi	Messaggio sospetto + click "Crea PDF"	Creazione file PDF con risultati analisi e consigli	Superato

ID	Nome Test Case	Input	Output Atteso	Stato
TC13	Controllo contenuto PDF	Apertura PDF generato	Riepilogo, livelli di rischio, raccomandazioni presenti	Superato
TC14	Verifica download PDF	Click su “Scarica PDF”	Avvio del download del documento	Superato

Tabella 33:TS04

Esempio di Test Suite: ANALISI NUMERI SOSPETTI (TS05)

ID	Nome Test Case	Input	Output Atteso	Stato
TC15	Identificazione numero spam	Numero Telefonico	Esito: “Numero sospetto”	Superato
TC16	Identificazione numero affidabile	Numero ufficiale	Nessuna Anomalia	Superato
TC17	Gestione formato numero non valido	Input non numerico o scorretto ("abc123")	Messaggio di errore: inserisci un numero valido con solo cifre e un prefisso selezionato.	Superato
TC18	Gestione gestore	Numero telefonico	Individuazione gestore	Superato
TC19	Identificazione numero fisso o mobile	Numero Telefonico	Individuazione Tipo	Superato
TC20	SQL Injection test	Input ' OR '1'='1 per prefisso e suffisso	Input bloccato	Superato

Tabella 34: TS05

Esempio di Test Suite: COMPILAZIONE FORM (TS06)

ID	Nome Test Case	Input	Output Atteso	Stato
TC21	Visualizzazione modulo	Accesso alla sezione “Rileva un attacco”	Il modulo viene caricato con domande semplici e riferimenti all’enciclopedia	Superato
TC22	Compilazione completa e invio corretto	Risposte fornite a tutte le domande, clic su “Invia risposte”	Il sistema elabora le risposte e fornisce un risultato (es. "Possibile phishing")	Superato
TC23	Invio con risposte parziali	Risposta mancante ad alcune domande	Il sistema blocca l’invio	Superato
TC24	Coerenza tra modulo e enciclopedia	Domande riferite all’attacco “Phishing”	Le domande rispecchiano i sintomi descritti nella relativa scheda dell’enciclopedia	Superato

ID	Nome Test Case	Input	Output Atteso	Stato
TC25	Accessibilità linguistica	Lettura e comprensione del modulo da parte di utente base	Linguaggio semplice e comprensibile	Superato

Tabella 35: TS06

La test suite proposta copre in modo completo e strutturato tutte le principali funzionalità della piattaforma CyberDefender, garantendo una verifica sistematica del corretto funzionamento del sistema sia dal punto di vista tecnico che dell'esperienza utente. I test sono suddivisi per area funzionale (encyclopedia, analizzatore, compilazione form, report PDF, verifica numeri) e formulati per validare comportamenti attesi in scenari sia standard che critici.

L'esecuzione dei test permetterà di:

- Verificare la correttezza e completezza dei contenuti informativi;
- Valutare l'accuratezza e la tempestività dell'analizzatore di minacce;
- Assicurare la generazione e distribuzione corretta dei report personalizzati;
- Garantire l'affidabilità della funzione di verifica numeri di telefono.

Una volta completata la fase di test, i risultati ottenuti consentiranno di identificare eventuali anomalie, ottimizzare l'esperienza utente e rafforzare l'efficacia generale della piattaforma nel campo della sensibilizzazione e prevenzione delle minacce informatiche.

Strategie di selezione dei test

Nel processo di verifica della qualità di un sistema complesso come CyberDefender, è fondamentale adottare strategie di testing mirate, capaci di garantire copertura funzionale, efficienza e robustezza. A tal fine, sono state impiegate tecniche consolidate come l'analisi delle classi di equivalenza, la valutazione dei valori limite e la selezione basata sul rischio.

L'analisi delle classi di equivalenza consente di ridurre il numero di casi di test raggruppando input simili, garantendo così una copertura rappresentativa con un numero contenuto di test. L'identificazione dei valori estremi, invece, permette di validare la tenuta del sistema in condizioni limite, spesso causa di bug nascosti. Infine, la selezione basata sul rischio concentra gli sforzi di testing sulle funzionalità più critiche, come l'analisi dei messaggi sospetti o la generazione dei report, assicurando maggiore attenzione ai possibili impatti negativi.

Classi di equivalenza

L'analisi delle classi di equivalenza è una tecnica di testing che permette di ridurre il numero di casi di test, suddividendo i possibili input in gruppi (classi) che si presume siano gestiti allo stesso modo dal sistema. Questo consente di scegliere pochi rappresentanti per ciascuna classe, migliorando l'efficienza dei test.

Si propone un'analisi delle classi di equivalenza per ciascuna delle funzionalità della piattaforma CyberDefender.

1. Enciclopedia degli attacchi

Parametro	Classi di Equivalenza	Valori Esempio
Tipo di attacco selezionato	✓ Attacco noto ✗ Attacco non presente ✗ Input vuoto o nullo	✓ "Phishing" ✗ "AttaccoXYZ" ✗ ""
Completezza della scheda	✓ Tutte le sezioni presenti ✗ Sezioni mancanti	✓ ogni categoria ha la lista di attacchi ✗ nome della categoria senza lista attacchi
Livello di comprensibilità	✓ Linguaggio semplice ✗ Termini tecnici non spiegati	✓ "In parole semplici..." ✗ "Exploit buffer overflow"

Tabella 36: Classi d'equivalenza1

2. Analizzatore di messaggi sospetti

Parametro	Classi di Equivalenza	Valori Esempio
Tipo di contenuto	✓ Testo benigno ✓ Testo sospetto ✗ Input vuoto	✓ "Ciao, ci vediamo" ✓ "Clicca qui per vincere" ✗ ""
Presenza di pattern noti	✓ Pattern rilevato ✗ Pattern non rilevato	✓ <script> ✗ "Buona giornata"
Formato del messaggio	✓ Testo semplice ✓ Codice ✗ Simboli casuali o incomprensibili	✓ "Gentile cliente..." ✓ DROP TABLE ✗ "@@!!\$%?"
Tipo di rischio assegnato	✓ Rischio basso ✓ Rischio medio ✓ Rischio alto	✓ testo neutro ✓ codice sospetto ✓ phishing diretto

Tabella 37: Classi d'equivalenza2

3. Compilazione modulo rilevamento attacco

Parametro	Classi di Equivalenza	Valori Esempio
Stato del modulo	✓ Completato correttamente ✗ Parzialmente completato ✗ Non compilato	✓ Tutte le risposte date ✗ 2 mancanti ✗ nessuna risposta
Rilevanza delle risposte	✓ Risposte coerenti con sintomi noti ✗ Risposte generiche o incoerenti	✓ “Si” ✗ “Non so”
Tipo di attacco selezionato	✓ Attacco presente in enciclopedia ✗ Attacco non classificabile	✓ Phishing ✗ “Attacco personalizzato non definito”
Comprensibilità linguistica	✓ Linguaggio semplice ✗ Linguaggio tecnico o ambiguo	✓ “Hai ricevuto email sospette che richiedono dati personali?” ✗ “Hai riscontrato payload”
Risultato del modulo	✓ Minaccia rilevata ✓ Nessun rischio ✗ Risultato non chiaro	✓ “Attacco: Rootkit (Malware e tecniche avanzate)” ✓ “Nessun attacco rilevato” ✗ “Errore elaborazione”

Tabella 38: Classi d'equivalenza3

4. Report PDF con consigli su misura

Parametro	Classi di Equivalenza	Valori Esempio
Esito dell'analisi	✓ Nessuna minaccia ✓ Una o più minacce rilevate	✓ Nessun attacco rilevato ✓ phishing + XSS
Stato del report	✓ Generato correttamente ✗ Report mancante o incompleto	✓ PDF completo ✗ PDF vuoto

Tabella 39: Classi d'equivalenza4

5. Analisi numeri sospetti

Parametro	Classi di Equivalenza	Valori Esempio
Tipo di numero	✓ Numero affidabile ✓ Numero sospetto ✗ Numero sconosciuto	✓ 800123456 ✓ 042156999 ✗ 3456789012
Formato del numero	✓ Numero valido ✗ Formato errato ✗ Caratteri non numerici	✓ 3331234567 ✗ 123 9864567 ✗ "abc123"
Risposta del sistema	✓ Numero classificato correttamente ✗ Nessuna risposta o errore	✓ " <input checked="" type="checkbox"/> Nessuna anomalia rilevata" ✗ errore 500

Tabella 40: Classi d'equivalenza5

L'analisi svolta offre numerosi vantaggi in fase di progettazione e ottimizzazione dei test:

- **Riduce il numero complessivo di test** necessari, poiché consente di rappresentare ogni classe con un singolo caso rappresentativo, evitando ripetizioni inutili.
- **Aumenta la copertura funzionale**, garantendo che tutte le tipologie rilevanti di input (validi, non validi, borderline) vengano considerate.
- **Facilita la pianificazione di test positivi e negativi**, permettendo di validare sia il comportamento atteso del sistema che la sua capacità di gestire errori o condizioni anomale in modo controllato.

Analisi dei valori limite

Sono stati definiti test specifici per i **limiti minimi e massimi previsti dal sistema**, al fine di verificare il corretto comportamento in condizioni estreme. In particolare:

- **Lunghezza minima e massima dei campi di input** (es. 0 caratteri, 255 caratteri).
- **Numero minimo e massimo di elementi selezionabili o inseribili** (es. 0 risposte in un modulo, 10 risposte totali).
- **Formati accettati e non accettati** (es. numero di telefono con 10 cifre corrette, formati troppo lunghi o con caratteri non validi).
- **Generazione report con contenuti minimi o molto estesi** (es. nessuna vulnerabilità rilevata, molte vulnerabilità).
- **Tempo minimo e massimo di risposta del sistema** (es. risposta istantanea, oltre il timeout accettabile).
- **Numero di richieste consecutive** (es. numero massimo di invii modulo o analisi prima di blocco temporaneo).

Questo approccio è fondamentale per individuare bug nascosti o comportamenti anomali che emergono solo in presenza di input al limite delle specifiche funzionali, garantendo così una maggiore robustezza e affidabilità del sistema.

Selezione basata su rischio

L'approccio adottato ha privilegiato i test relativi alle funzionalità più sensibili e soggette a potenziali impatti negativi in caso di malfunzionamento:

- **Analisi dei messaggi sospetti:** valutazione dell'efficacia nel rilevare contenuti malevoli, gestione di input complessi o borderline.
- **Modulo per il rilevamento di attacchi:** verifica della pertinenza tra i sintomi descritti e la tipologia di attacco rilevato.
- **Verifica dei numeri di telefono:** accuratezza nel distinguere numeri legittimi da contatti potenzialmente fraudolenti.
- **Generazione di report PDF personalizzati:** test delle funzionalità di esportazione in presenza di dati minimi, medi e complessi.
- **Encyclopedia degli attacchi:** controllo dei contenuti per le tipologie di minaccia più comuni e critiche per l'utente.

Esempio applicativo

Per la funzionalità “**Analizzatore di messaggi sospetti**”, i test selezionati includono:

- **Caso standard:** messaggio contenente termini dubbi, ma non dannosi, per valutare la sensibilità dell'analisi.
- **Estremo superiore:** messaggio molto lungo, con più elementi a rischio (link sospetti, script, testo offuscato).
- **Input anomalo:** messaggio vuoto o contenente codice in formato irregolare (es. base64 misto a JavaScript).

Questo metodo ha consentito di focalizzare i test sulle aree a maggiore impatto, bilanciando efficacia e copertura. Il risultato è stato lo sviluppo di una suite di test mirata e sostenibile, in grado di ottimizzare il processo di validazione senza compromettere la qualità.

Framework e strumenti per i test

Per assicurare un processo di verifica strutturato e affidabile, sono stati adottati strumenti e framework mirati a coprire ogni livello del ciclo di testing. L'approccio scelto punta a validare non solo il corretto funzionamento del software, ma anche aspetti fondamentali come la sicurezza, le prestazioni e la qualità del codice, con un focus particolare sulla prevenzione delle vulnerabilità informatiche.

Framework per test unitari e di integrazione

Unittest: è il framework ufficiale di testing fornito con Python. È utilizzato per scrivere test automatizzati che verificano il corretto funzionamento del codice, rilevando bug e regressioni.

Funzionamento:

- Si definisce una classe di test che eredita da unittest.TestCase.
- Ogni metodo che comincia con test_ è considerato un caso di test.
- Si usano metodi di asserzione come assertEquals(), assertTrue(), assertRaises(), ecc.

Vulnerability Scanning

sqlmap: Strumento automatico per il rilevamento di vulnerabilità da **SQL injection**. Utilizzato in fase di test del backend per assicurarsi che le interazioni con il database, specialmente quelle provenienti da input utente (es. numeri di telefono o messaggi sospetti), siano adeguatamente sanificate.

```
[17:55:06] [WARNING] you've provided target URL without any GET parameters (e.g. 'http://www.site.com/article.php?id=1') and without providing any POST parameters through option '--data'
do you want to try URI injections in the target URL itself? [Y/n/q]

[17:55:08] [INFO] testing connection to the target URL
you have not declared cookie(s), while server wants to set its own ('csrfToken=EIbQRmxk5RT...890W3Ku4zX'). Do you want to use those [Y/n]

[17:55:09] [INFO] testing if the target URL content is stable
[17:55:09] [WARNING] target URL content is not stable (i.e. content differs). sqlmap will base the page comparison on a sequence matcher. If no dynamic nor injectable parameters are detected, or in case of junk results, refer to user's manual paragraph 'Page comparison'
how do you want to proceed? [(C)ontinue/(S)tring/(R)egex/(Q)uit]

[17:55:10] [INFO] testing if URI parameter '#1*' is dynamic
[17:55:10] [WARNING] URI parameter '#1*' does not appear to be dynamic
[17:55:10] [WARNING] heuristic (basic) test shows that URI parameter '#1*' might not be injectable
[17:55:10] [INFO] testing for SQL injection on URI parameter '#1*'
[17:55:10] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[17:55:10] [WARNING] reflective value(s) found and filtering out
[17:55:11] [INFO] testing 'Boolean-based blind - Parameter replace (original value)'
[17:55:11] [INFO] testing 'MySQL >= 5.1 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (EXTRACTVALUE)'
[17:55:11] [INFO] testing 'PostgreSQL AND error-based - WHERE or HAVING clause'
[17:55:12] [INFO] testing 'Microsoft SQL Server/Sybase AND error-based - WHERE or HAVING clause (IN)'
[17:55:12] [INFO] testing 'Oracle AND error-based - WHERE or HAVING clause (XMLType)'
[17:55:13] [INFO] testing 'Generic inline queries'
[17:55:13] [INFO] testing 'PostgreSQL > 8.1 stacked queries (comment)'
[17:55:13] [INFO] testing 'Microsoft SQL Server/Sybase stacked queries (comment)'
[17:55:13] [INFO] testing 'Oracle stacked queries (DBMS_PIPE.RECEIVE_MESSAGE - comment)'
[17:55:14] [INFO] testing 'MySQL >= 5.0.12 AND time-based blind (query SLEEP)'
[17:55:14] [INFO] testing 'PostgreSQL > 8.1 AND time-based blind'
[17:55:14] [INFO] testing 'Microsoft SQL Server/Sybase time-based blind (IF)'
[17:55:15] [INFO] testing 'Oracle AND time-based blind'
it is recommended to perform only basic UNION tests if there is not at least one other (potential) technique found. Do you want to reduce the number of requests? [Y/n]

[17:55:15] [INFO] testing 'Generic UNION query (NULL) - 1 to 10 columns'
[17:55:16] [WARNING] URI parameter '#1*' does not seem to be injectable
[17:55:16] [CRITICAL] all tested parameters do not appear to be injectable. Try to increase values for '--level'/'--risk' options if you wish to perform more tests. If you suspect that there is some kind of protection mechanism involved (e.g. WAF) maybe you could try to use option '--tamper' (e.g. '--tamper=space2comment') and/or switch '--random-agent'
[17:55:16] [WARNING] HTTP error codes detected during run:
404 (Not Found) - 73 times
```

Com'è osservabile non è stato possibile iniettare nessun input malevolo nella pagina dedicata al login dell'utente, che abbia portato a termine un attacco di tipo SQL Injection. Sqlmap ha testato molte tecniche di iniezione (boolean, error-based, time-based, union, stacked queries) ma nessuna ha funzionato. Ha concluso che i parametri testati non sono vulnerabili a SQL injection classiche. Il server ha risposto "pagina non trovata" per quasi tutte le richieste.

```

Not Found: /login/) ORDER BY 1-- JbCF
[WARNING] Not Found: /login/) ORDER BY 1-- JbCF
[24/Jul/2025 16:41:14] "GET /login/%29%20ORDER%20BY%201--%20JbCF HTTP/1.1" 404 5692
Not Found: /login/ ORDER BY 1-- nRwj
[WARNING] Not Found: /login/ ORDER BY 1-- nRwj
[24/Jul/2025 16:41:14] "GET /login/%20ORDER%20BY%201--%20nRwj HTTP/1.1" 404 5689
Not Found: /login/') ORDER BY 1-- D0tv
[WARNING] Not Found: /login/') ORDER BY 1-- D0tv
[24/Jul/2025 16:41:14] "GET /login/%27%29%20ORDER%20BY%201--%20D0tv HTTP/1.1" 404 5710
Not Found: /login/' ORDER BY 1-- ajZI
[WARNING] Not Found: /login/' ORDER BY 1-- ajZI

```

Dalla console di Django si vede che tutti gli url con payload richiesti con sqlmap vengono trasformati in stringhe dall'ORM di Django in automatico, rendendo nulla la possibilità di SQL Injection.

```

[18:21:48] [WARNING] you've provided target URL without any GET parameters (e.g. 'http://www.site.com/article.php?id=1') and without providing any POST parameters through option '--data'
do you want to try URI injections in the target URL itself? [Y/n/q]

[18:21:56] [INFO] testing connection to the target URL
[18:21:56] [INFO] checking if the target is protected by some kind of WAF/IPS
[18:21:56] [INFO] testing if the target URL content is stable
[18:21:57] [INFO] target URL content is stable
[18:21:57] [INFO] testing if URI parameter '#1*' is dynamic
[18:21:57] [WARNING] URI parameter '#1*' does not appear to be dynamic
[18:21:57] [WARNING] heuristic (basic) test shows that URI parameter '#1*' might not be injectable
[18:21:57] [INFO] testing for SQL injection on URI parameter '#1*'
[18:21:57] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[18:21:57] [WARNING] reflective value(s) found and filtering out
[18:21:58] [INFO] testing 'Boolean-based blind - Parameter replace (original value)'
[18:21:58] [INFO] testing 'MySQL >= 5.1 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (EXTRACTVALUE)'
[18:21:58] [INFO] testing 'PostgreSQL AND error-based - WHERE or HAVING clause'
[18:21:58] [INFO] testing 'Microsoft SQL Server/Sybase AND error-based - WHERE or HAVING clause (IN)'
[18:21:58] [INFO] testing 'Oracle AND error-based - WHERE or HAVING clause (XMLType)'
[18:21:58] [INFO] testing 'Generic inline queries'
[18:21:58] [INFO] testing 'PostgreSQL > 8.1 stacked queries (comment)'
[18:21:59] [INFO] testing 'Microsoft SQL Server/Sybase stacked queries (comment)'
[18:21:59] [INFO] testing 'Oracle stacked queries (DBMS_PIPE.RECEIVE_MESSAGE - comment)'
[18:21:59] [INFO] testing 'MySQL >= 5.0.12 AND time-based blind (query SLEEP)'
[18:21:59] [INFO] testing 'PostgreSQL > 8.1 AND time-based blind'
[18:21:59] [INFO] testing 'Microsoft SQL Server/Sybase time-based blind (IF)'
[18:21:59] [INFO] testing 'Oracle AND time-based blind'
it is recommended to perform only basic UNION tests if there is not at least one other (potential) technique found. Do you want to reduce the number of requests? [Y/n]

[18:22:00] [INFO] testing 'Generic UNION query (NULL) - 1 to 10 columns'
[18:22:00] [WARNING] URI parameter '#1*' does not seem to be injectable
[18:22:00] [CRITICAL] all tested parameters do not appear to be injectable. Try to increase values for '--level'/'--risk' options if you wish to perform more tests. If you suspect that there is some kind of protection mechanism involved (e.g. WAF) maybe you could try to use option '--tamper' (e.g. '--tamper=safe2comment') and/or switch '--random-agent'.
[18:22:00] [WARNING] HTTP error codes detected during run:
404 (Not Found) - 73 times

[*] ending @ 18:22:00 /2025-07-24/

```

Avviato sqlmap, passando un URL che non conteneva parametri GET e senza specificare nemmeno dei dati POST tramite l'opzione --data. In assenza di parametri chiari da testare, sqlmap ha tentato un approccio chiamato URI injection, cioè ha provato a iniettare codice direttamente nella struttura del percorso dell'URL (es. /pagina). Durante il test, sqlmap ha analizzato un parametro隐式 del percorso, identificato come #1*, ma ha rilevato che questo non era dinamico, quindi non vulnerabile a SQL injection. Nonostante ciò, ha comunque provato tutte le principali tecniche di iniezione SQL (come quelle basate su condizioni booleane, errori e ritardi temporali), ma nessuna di queste ha avuto successo. Inoltre, molte delle richieste inviate al server hanno restituito errori 404, ovvero "pagina non trovata", il che indica che molti tentativi di iniezione hanno colpito URL inesistenti.

Static Code Analysis

- **Bandit:** Utilizzato per esaminare automaticamente il codice Python alla ricerca di pattern insicuri o potenziali vulnerabilità.

Test Non Funzionali: Performance e scalabilità

Oltre ai test funzionali, sono stati eseguiti test non funzionali volti a valutare le prestazioni della piattaforma CyberDefender e la sua capacità di gestire un numero crescente di richieste e dati, assicurando reattività e affidabilità in situazioni di utilizzo reale.

Test di Performance

I test si sono concentrati sulle operazioni chiave della piattaforma, fondamentali per garantire un'esperienza utente fluida e sicura:

- **Analisi di messaggi sospetti:** misurazione dei tempi necessari per processare input testuali di diverse dimensioni e complessità, inclusi messaggi contenenti pattern sospetti o codice offuscato.
- **Generazione report PDF personalizzati:** verifica della velocità nel creare documenti che riassumono i risultati dell'analisi, con contenuti di lunghezza variabile e dettagli di sicurezza.
- **Verifica numeri di telefono sospetti:** test dei tempi di risposta nell'autenticare e classificare numeri di telefono come affidabili o fraudolenti.
- **Compilazione modulo di rilevamento attacco:** valutazione della reattività del sistema nella gestione delle risposte utente durante il completamento guidato del modulo.

Strumenti utilizzati

time (Unittest): indica il tempo impiegato per eseguire i test scritti con il framework unittest (modulo di test integrato in Python).

Obiettivi e risultati attesi

- Tempo di analisi dei messaggi sospetti inferiore a 3 secondi per input di medie dimensioni (fino a 5.000 caratteri).

```
def test_response_time_under_3_seconds(self): new*  
    start_time = time.time()  
    response = self.client.post(reverse('checkMessage'), data: {'text': 'Messaggio urgente: aggiorna la password!'}  
    duration = time.time() - start_time  
    self.assertLessEqual(duration, b: 3, msg: f"Tempo di risposta troppo lungo: {duration:.2f} secondi")  
    self.assertEqual(response.status_code, second: 200)
```

Test Suite

La suite include test di carico che simulano più utenti contemporanei che utilizzano le funzionalità critiche, come l'inserimento e l'analisi simultanea di messaggi sospetti o la verifica parallela di numeri telefonici, per valutare la scalabilità e la stabilità della piattaforma sotto stress.

Test Suite: AUTENTICAZIONE UTENTE (TS01)

```
(.venv) PS C:\Users\filom\PycharmProjects\DjangoProject\SW_Sicuro_Website\website> python manage.py test
Found 5 test(s).
Creating test database for alias 'default'...
System check identified no issues (0 silenced).

.....
-----
Ran 5 tests in 1.714s

OK
Ran 5 tests in 1.714s
```

Esempio di Test Suite: ANALISI MESSAGGI SOSPETTI (TS02)

```
(venv) PS C:\Users\filom\PycharmProjects\DjangoProject\SW_Sicuro_Website\website> python manage.py test analizzatoreMessaggi.tests
Found 6 test(s).
Creating test database for alias 'default'...
System check identified no issues (0 silenced).

.....
-----
Ran 6 tests in 0.027s

OK
Destroying test database for alias 'default'...
(venv) PS C:\Users\filom\PycharmProjects\DjangoProject\SW_Sicuro_Website\website>
```

Test Suite: ENCICLOPEDIA DEGLI ATTACCHI (TS03)

```
(.venv) PS C:\Users\filom\PycharmProjects\DjangoProject\SW_Sicuro_Website\website> python manage.py test enciclopedia.tests
Found 4 test(s).
Creating test database for alias 'default'...
System check identified no issues (0 silenced).
Consultazione salvata per utente: test@example.com
.Nessun utente loggato in sessione
.[WARNING] Not Found: /enciclopedia_attacchi/9999/
.Nessun utente loggato in sessione
.

.....
Ran 4 tests in 0.066s

OK
```

Test Suite: REPORT PDF CONSIGLI SU MISURA (TS04)

```
(.venv) PS C:\Users\filom\PycharmProjects\DjangoProject\SW_Sicuro_Website\website> python manage.py test enciclopedia.tests
Found 1 test(s).
Creating test database for alias 'default'...
System check identified no issues (0 silenced).

.

.....
Ran 1 test in 0.432s

OK
```

Test Suite: ANALISI NUMERI SOSPETTI (TS05)

```
(.venv) PS C:\Users\filom\PycharmProjects\DjangoProject\SW_Sicuro_Website\website> python manage.py test analisi_numeri.tests
Found 5 test(s).
Creating test database for alias 'default'...
System check identified no issues (0 silenced).
.....
-----
Ran 5 tests in 0.087s

OK
```

Funzione per validare input con prefisso invalido:

```
def test_sql_injection_attempt_prefisso(self): new *
    """Tentativo SQL Injection via prefisso"""
    response = self.client.post(self.url, data={
        'prefisso': "' OR '1'='1",
        'numero': "3211234567"
    })

    self.assertEqual(response.status_code, 200)
    self.assertContains(response, text="Inserisci un numero valido con solo cifre e un prefisso selezionato.")

    self.assertEqual(NumerosTelefonico.objects.count(), 0)
```

Funzione che non valida il prefisso malevolo:

```
def test_sql_injection_attempt_prefisso(self): new *
    """Tentativo SQL Injection via prefisso"""
    response = self.client.post(self.url, {
        'prefisso': "+39' OR '1'='1",
        'numero': "3211234567"
    })

    self.assertEqual(response.status_code, 200)
    self.assertContains(response, "Inserisci un numero valido con solo cifre e un prefisso selezionato.")
    self.assertEqual(NumerosTelefonico.objects.count(), 0)
```

Test Suite: COMPILAZIONE FORM (TS06)

```
(venv) PS C:\Users\filom\PycharmProjects\DjangoProject\SW_Sicuro_Website\website> python manage.py test enciclopedia.tests
Found 6 test(s).
Creating test database for alias 'default'...
System check identified no issues (0 silenced).
..Ricerca attacco: Test Attacco
.....
-----
Ran 6 tests in 0.575s

OK
```

Sommario

1. Cos'è CyberDefender?	3
Descrizione generale del progetto	3
Target degli utenti	3
Funzionalità Principali.....	4
Product Vision	5
Elevator Pitch	5
Obiettivi principali.....	5
Requisiti generali dettagliati	6
2. Organizzazione Agile e Sprint.....	7
Metodologia Agile adottata.....	7
Panoramica Sprint	7
Dettaglio Sprint.....	8
Conclusione Sprint con ROADMAP	8
3. Analisi dei requisiti.....	10
Tecniche di Elicitazione dei Requisiti	10
Classificazione dei Requisiti.....	11
Personas – Profili Rappresentativi.....	14
Casi d'Uso.....	15
User Stories.....	20
Matrice di Tracciabilità dei Requisiti	21
Misuse Case: Rischi e Contromisure	23
Requisiti di privacy e compliance	25
4. Analisi delle minacce (Threat Modeling).....	28
Obiettivo e metodo	28
Data Flow Diagram (DFD) - level 0.....	28
Legenda colori	29
DFD level 1 - Sprint 1 – Consultazione dell'enciclopedia attacchi	32
DFD level 1 - Sprint 2 – Implementazione login e analisi natura messaggi e numeri	33
DFD level 1 - Sprint 3 – Implementazione form e report PDF.....	34
STRIDE – Classificazione delle minacce.....	35
Minacce specifiche identificate	35
Mitigazioni generali	39
5. Progettazione dell'architettura sicura.....	41
Pattern architetturali adottati	41
Principi di Progettazione Sicura Applicativa	43

Vista Architetturale UML	43
Modello E-R e Logico del Database.....	49
6. Sviluppo sicuro del software	53
Scelte tecnologiche e linguaggio	53
Best Practices di programmazione sicura	54
Gestione degli errori ed eccezioni.....	54
Autenticazioni e autorizzazioni	56
Rate Limiting e durata massima sessione	57
Gestione sicura della cifratura delle password.....	58
Verifica di sicurezza automatica	58
7. Testing e verifica del software	60
Obiettivi del testing	60
Metodologie Adottate.....	60
Tipologie di test.....	61
Test Misuse Case.....	61
Organizzazione dei test: Test Suite e Test Case	62
Strategie di selezione dei test	65
Classi di equivalenza	65
Analisi dei valori limite	68
Selezione basata su rischio	69
Framework e strumenti per i test	69
Framework per test unitari e di integrazione	70
Vulnerability Scanning	70
Test Non Funzionali: Performance e scalabilità.....	72
Test di Performance.....	72
Strumenti utilizzati.....	72
Obiettivi e risultati attesi	72
Test Suite	73