TECHNICAL REPORT

Indice

1. Introduzione

Questo report descrive le vulnerabilità individuate nell'applicazione web "GeekFactory". I risultati presentati in questo report tecnico non rappresentano necessariamente una dichiarazione esaustiva di tutte le vulnerabilità e criticità esistenti. È quindi possibile che esistano o sorgano vulnerabilità non identificate durante la nostra revisione. Quello che viene richiesto è risolvere soltanto le vulnerabilità descritte in questo report.

2. Metodologia di analisi

Le vulnerabilità sono state individuate attraverso tecniche di analisi dinamica e statica.

- L'**analisi dinamica** ha identificato comportamenti insicuri dell'applicazione in esecuzione. Per tali vulnerabilità, viene indicata la proof of concept, cioè come replicare il comportamento insicuro durante l'esecuzione dell'applicazione. **Nota bene**: non essendo evidenziati i file e le linee di codice vulnerabili, la risoluzione di tali vulnerabilità richiede allo sviluppatore di identificare sia i file che le righe di codice da modificare.

- L'**analisi statica** del codice sorgente ha esaminato l'intero codice sorgente dell'applicazione alla ricerca di vulnerabilità. Per tali vulnerabilità, vengono indicate i file e le linee di codice in cui sono state individuate. **Nota bene**: la risoluzione di tali vulnerabilità *potrebbe* richiedere la modifica di *altre* righe di codice.

Ciascuna vulnerabilità individuata è stata classificata secondo i rischi di sicurezza della OWASP Top 10 (versione 2021) e con il CVSS Score. In accordo al CVSS Score, alle vulnerabilità presentate in questo documento sarà assegnata una delle seguenti livelli di severity:

Livelli di severity	Descrizione	Range CVSS
Critical	La vulnerabilità permette all'attaccante di compromettere del tutto l'asset.	9.0 - 10.0
High	La vulnerabilità consente l'esecuzione di codice malevolo, ma non permette l'accesso amministrativo alle risorse o ai dati presenti nel database.	7.0 - 8.9
Medium	La vulnerabilità consente di acquisire informazioni per successivi attacchi. Potrebbe inoltre permettere all'attaccante di modificare le normali operazioni dell'asset.	4.0 - 6.9
Low	La vulnerabilità consente di recuperare informazioni sulle attività del cliente, ma con un impatto molto basso sulle attività.	0.1 - 3.9
Informational	La vulnerabilità è presente ma non è stato possibile sfruttarla durante l'attività di security assessment.	0

3. Il perimetro di analisi

Il perimetro di analisi è descritto dalla seguente tabella:

Applicazione	Ambiente	URL
GeekFactory	Test	http://localhost:8080/GeekFactory2/

L'applicazione "GeekFactory" è stata analizzata alla ricerca di vulnerabilità in ambiente di test. L'applicazione web è stata hostata in *localhost* sulla *porta 8080*.
Nota bene: rieseguire l'applicazione sul proprio computer (e con le proprie configurazioni per il web server) potrebbe prevedere un URL diverso (es. un diverso numero di porta).

Per avviare l'applicazione web, si suggerisce di utilizzare:
- Eclipse IDE for Enterprise Java and Web Developers - 2021-12
- Apache Tomcat v9.0 Server at localhost
- JavaSE-17

Inoltre, si consiglia di utilizzare MySQL come Database Management System. È possibile utilizzare lo script "creazione_e_popolamento_database.sql" per creare e popolare il database che verrà utilizzato dall'applicazione web. Tale script si trova nella cartella "database".

Così come riportato all'interno dello script "creazione_e_popolamento_database.sql", è possibile utilizzare le seguenti credenziali per accedere come *cliente*:
- Username/email: mariorossi@gmail.com
- Password: 12345678

Invece, per accedere come *admin*, è possibile utilizzare le seguenti credenziali:
- Username/email: geekfactory@gmail.com
- Password: 12345678

Nota bene: le credenziali per la connessione al database sono:
- Username: root
- Password: root

In caso fosse necessario cambiarle (per accedere al database), è possibile modificarle nel file *src/main/java/model/DriverManagerConnectionPool.java* (riga 26 e riga 27, rispettivamente).

Nota bene: per modificare la password di un utente di MySQL, è possibile utilizzare una query SQL; tale query SQL, per l'utente avente come username "root", è la seguente:

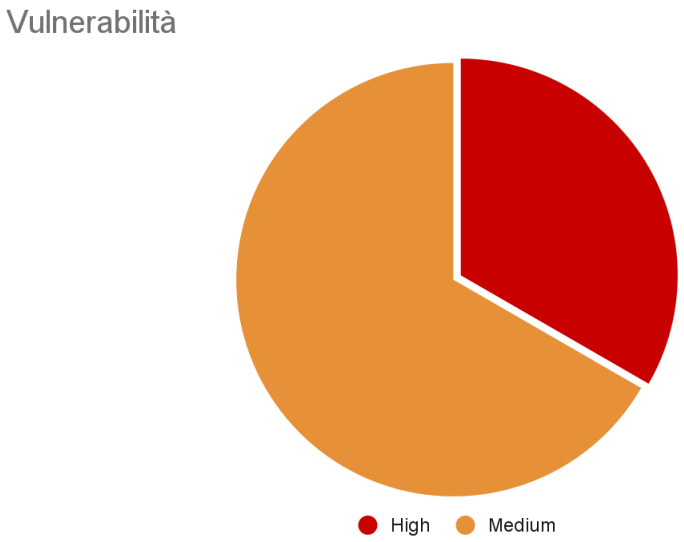
SET PASSWORD FOR 'root'@'localhost' = 'root';

Nota bene: rieseguire l'applicazione sul proprio computer (e con le proprie configurazioni per il web server) potrebbe prevedere un URL diverso (es. un diverso numero di porta).

Nota bene: i file web.xml, JSP, JavaScript, CSS, etc. si trovano sotto la cartella *src/main/webapp*, la quale è analoga alla cartella *WebContent*.

4. Riepilogo

Questa sezione riepiloga le vulnerabilità individuate nell'applicazione.



L'attività svolta ha evidenziato la presenza di 6 vulnerabilità, distribuite in questo modo:

- **2** vulnerabilità ad impatto **high**;
- **4** vulnerabilità ad impatto **medium**.

Il livello viene determinato in accordo agli standard de facto internazionali (CVSS) per la sicurezza delle informazioni, tenendo conto delle vulnerabilità che potrebbero compromettere l'integrità e la riservatezza dell'applicazione e dei dati in essa contenuti.

4.1 Istruzioni per affrontare le vulnerabilità

- Le seguenti sezioni di questo documentano illustrano le vulnerabilità che è necessario individuare e risolvere all'interno dell'applicazione web.
- Per ciascuna vulnerabilità, vengono riportate le seguenti informazioni:
    - Nome
    - CVSS Score
    - Classificazione in accordo alla OWASP Top 10
    - Status
    - Descrizione, la quale serve a fornire informazioni sul problema riscontrato
    - *Proof of concept* o la *localizzazione*, a seconda se la vulnerabilità è stata individuata attraverso analisi dinamica o statica dell'applicazione web
    - La remediation, la quale suggerisce strategie per la risoluzione della vulnerabilità
    - Riferimenti alla classificazione CWE
- Per mettere in sicurezza l'applicazione web, si richiede di leggere la documentazione fornita in questo technical report e di individuare e risolvere le vulnerabilità.

- **È OBBLIGATORIO individuare e risolvere ciascuna vulnerabilità nell'ordine in cui vengono presentate nel technical report.** Questo significa che dovrete affrontare prima della vulnerabilità avente ID-1, poi ID-2, etc.
- **Quando avete finito di affrontare una vulnerabilità, DOVETE effettuare un commit per comunicarlo!**
- **Questo deve essere fatto sia in caso avete risolto la vulnerabilità che altrimenti**. Ad esempio, potreste decidere di non continuare ad affrontare una certa vulnerabilità e di voler passare alla successiva.
- **Prima di passare ad una nuova vulnerabilità, è NECESSARIO effettuare un commit per dire che avete finito di affrontare la vulnerabilità che stavate affrontando.** Questo significa che iniziate ad affrontare prima la vulnerabilità avente ID-1; poi, quando avete finito, EFFETTUATE UN COMMIT e DOPO passate alla vulnerabilità ID-2 (e così via per tutte le altre vulnerabilità).
- **Il messaggio di commit DEVE avere la seguente struttura:**
    - Nel caso in cui pensate di aver risolto la vulnerabilità: ***id-1 resolved***
        - Sostituite *id-1* con l'id della vulnerabilità che avete affrontato (id-1 è per la prima vulnerabilità, id-2 è per la seconda vulnerabilità, etc.)
    - Nel caso in cui non avete risolto la vulnerabilità: ***id-1 unresolved***
        - Sostituite *id-1* con l'id della vulnerabilità che avete affrontato (id-1 è per la prima vulnerabilità, id-2 è per la seconda vulnerabilità, etc.)
        - Se non avete modificato alcun file, potreste ritrovarvi nella condizione in cui non riuscite ad effettuare il commit per dire che non avete risolto la vulnerabilità (non ci riuscite perché per effettuare un commit è necessario aver modificato almeno un file). In tal caso, aprite un file qualsiasi ed effettuate una modifica non significativa, come l'aggiunta di uno spazio; successivamente, effettuate il commit.