Report Flagify

Introduzione

Flagify è una piattaforma che nasce con l'intento di testare le proprie skill nella risoluzione di problemi di Cybersecurity di vario genere: *Web Security*, *Reverse Engineering, Cryptography*, *Binary Exploitation*, *Software Security*, e altre ancora.

Tutte le pagine html all'interno del sito utilizzano plugin di bootstrap e jQuery, mentre la maggior parte delle richieste lato Client vengono effettuate tramite XMLHttpRequest.

Homepage (static/templates/homepage)

La parte interessante da considerare nella nostra homepage è il form, fatto per permettere agli utenti, di interagire con coloro che mantengono il sito (cioè noi).

Compilando i tre campi di questo form (suo indirizzo email, oggetto della mail, corpo) l'utente può mandarci un'email: questo servizio è stato gestito tramite un'apposita funzione scritta in nodejs che sostanzialmente consente di "auto-inviare" un'email al nostro indirizzo flagify@outlook.it.

Un'altra funzione da considerare nella cartella homepage è quella nel file script.js che serve per avere in alto a destra il nome dell'utente attualmente loggato al posto dei pulsanti che si trovano quando si accede inizialmente al sito (Login/Signup).

Login/Signup (static/templates/login) / (static/templates/signup)

I pulsanti di Login e Signup indirizzano rispettivamente alle pagine di login, in cui un utente già registrato può accedere al proprio profilo, e di signup, in cui un utente nuovo si può registrare. Sia per il login che per il signup vengono gestite le varie casistiche di errore:
il caso in cui l'utente sbagli e-mail, password, o inserisca password che non corrispondono nel signup, mostrando una stringa di errore.

Challenges (static/templates/challenges)

Per le cards delle challenge vengono utilizzate varie funzioni in successione, le quali fanno richieste al server tramite XMLHttpRequest.
Vengono prese tutte le categorie di challenge e per ogni categoria crea un carosello, tramite il plugin owl-carousel, contenente tutte le challenge ad essa appartenenti. Dopo aver

disposto tutte le cards, colora di verde quelle già risolte dall'utente di sessione e viene preso il nome dell'utente e messo vicino al pulsante profilo.

Successivamente quando si clicca su una card, viene mostrato un modal bootstrap che prende dal server le informazioni della challenge da far vedere. All'interno di questo modal vi è un modal annidato per la richiesta di sblocco e per la visualizzazione degli hint.

Challenge (static/templates/challenge)

Nella cartella challenge vi sono i file di alcune challenge presenti sul sito.
In particolare, i file relativi alle varie challenge che si possono scaricare dal sito vengono presi da qui, passati come parametri in una richiesta di tipo **GET**, e possono essere file html a cui si viene reindirizzati o file di altro genere da scaricare.

Scoreboard (static/templates/scoreboard)

La scoreboard utilizza una table bootstrap che mostra la classifica degli utenti.
Dispone di un form-select da cui poter filtrare la scoreboard in base alle categorie presenti e di link profilo-utente dove è possibile cliccare sui nomi degli altri utenti per poter visualizzare i loro profili e le loro statistiche.

Profilo (static/templates/profile)

Il profilo, similmente alla scoreboard, utilizza varie table bootstrap che vengono popolate tramite funzioni JavaScript che richiedono informazioni al server riguardo l'utente e alle sue statistiche.

Sfondo (static/js/sfondo.js)

Lo sfondo utilizzato nella pagina delle challenge, situato in static/js/sfondo.js, è uno script js che permette di avere un background dinamico costituito da linee colorate che si muovono pseudo-randomicamente.

Server (server.js)

Il server del nostro sito è scritto in nodejs, ed è suddiviso in 6 sezioni:

- La prima sezione è riservata alle impostazioni per l'utilizzo del backend node e ad alcune funzioni come restrict() per permettere l'accesso ad alcune pagine solo agli utenti loggati.

- La seconda serve per prendere le informazioni relative alle challenge dal database, come il titolo, la descrizione, la flag e tanto altro.
- La terza si focalizza sul prendere le informazioni relative agli utenti dal database, come l'username, l'email e le statistiche calcolate tramite query apposite.
- La quarta sezione reindirizza l'utente alle path che vuole raggiungere.
  Es: /homepage o /challenges.
- La quinta è riservata alle funzioni post del login e del signup che comunicano direttamente col database e salvano le modifiche effettuate all'utente.
- L'ultima parte riguarda alcune funzioni ausiliarie, come quella dedicata all'invio di una email di feedback.

Database (db)

Il database è di tipo PostgreSQL ed è formato da tre tabelle, come mostrato nel file db/database_layout.txt:

- *utente*
- *challenge*
- *utente_challenge*

La prima è formata dalle informazioni base dell'utente e ha come chiave primaria l'username.
La seconda contiene informazioni riguardo ogni singola challenge ed ha come chiave primaria l'id della stessa.
La tabella *utente_challenge* ha una duplice chiave primaria formata dall'username di un utente e l'id di una challenge; in quest'ultima vengono salvate i timestamp della risoluzione della challenge e della richiesta dell'hint utilizzati da ogni singolo utente.

by
Thomas Kirschner
Nicolò Della Porta
Edoardo Giuggioloni