EvoCinema

“ Sistema per la gestione di un cinema “

Versione 1.0

Security and Recovery Test

Coordinatori del progetto

Prof. Andrea De Lucia - Top Manager
Francesco Vicidomini - Project Manager
Ferdinando D'Avino - Project Manager

Partecipanti

Luca Strefezza	0512102474	strluca94@gmail.com
Angelo Stefano D'Auria	0512102630	angelodauria91@gmail.com
Gianluca Villani	0512102990	lucassalerno1995@gmail.com
Giuseppe D'Ambrosio	0512103472	giuseppe.dambrosio14@gmail.com
Giuseppe Apuzzo	0512103920	g.apuzzo94@gmail.com
Sara De Filippo	0512103430	s.defilippo93@gmail.com
Antonio Giulio	0512103098	antonio.giulio96@gmail.com
Michele Delli Paoli	0512103820	mikeledellipaoli@gmail.com
Giuseppe Del Gaudio	0512103690	ciaogiuseppe96@gmail.com
Pietro Dell'Isola	0512103866	dellisola.pietro@gmail.com
Emanuele Buono	0512102370	squareman93@gmail.com
Francesco De Feo	0512103274	francescodefeo94@gmail.com

Revision history

Data	***Versione***	***Descrizione***	***Autori***
31/01/2018	1.0	Stesura del Security and Recovery Test	Antonio Giulio, Emanuele Buono, Giuseppe D'Ambrosio, Sara De Filippo

1. Introduzione

Il Security o Penetration test è il processo operativo di valutazione della sicurezza di un sistema o di una rete che simula l'attacco di un utente malintenzionato. L'analisi comprende più fasi ed ha come obiettivo evidenziare le debolezze della piattaforma fornendo il maggior numero di informazioni sulle vulnerabilità che ne hanno permesso l'accesso non autorizzato. L'analisi è condotta dal punto di vista di un potenziale attaccante e consiste nello sfruttamento delle vulnerabilità rilevate al fine di ottenere più informazioni possibili per accedere indebitamente al sistema.

2. Fasi

Nel caso di Evocinema, il Security test è stato suddiviso in 3 fasi:

- SQL Injection test
- JavaScript-HTML XSS test
- Privilege Escalation test

3. SQL Injection

Un SQL injection (SQLi) è un attacco mirato a colpire le applicazioni web che si appoggiano su un DBMS di tipo SQL. Questo attacco sfrutta l'inefficienza dei controlli sui dati ricevuti in input ed inserisce codice maligno all'interno di una query SQL. Le conseguenze prodotte sono imprevedibili per il programmatore, l'SQL injection permette al malintenzionato di autenticarsi con ampi privilegi in aree protette del sito anche senza essere in possesso delle credenziali di accesso e di visualizzare e/o alterare dati presenti del database.
Evocinema interagisce con l‘utente, che può inserire dei dati, e quindi potenzialmente effettuare una SQLi. La SQLi in se, deve contenere dei caratteri specifici della sintassi SQL, come ad esempio „ (l‟apostrofo), “ (gli apici), ; (punto e virgola) ecc... La verifica dell’esistenza di questi caratteri nell’ input, garantisce l’impossibilità di effettuare una iniezione.
Tutti i campi input di Evocinema, prima di essere inseriti nella query verso il db, vengono validati con dei pattern regex (ad esempio nome utente viene validato con /^[a-zA-Z0-9_ .]+$/, il quale rende impossibile l’inserimento dei caratteri necessari per una SQLi)
La validazione avviene nei due momenti diversi: lato client e lato server. Lato client non è sicuro, siccome un malintenzionato potrebbe eseguire una richiesta direttamente al server sorpassando validazione con jquery. La seconda verifica, lato server, impossibile sorpassarla, quindi rende il sistema sicuro. I campi, dove sono necessari i caratteri specifici, ad esempio una descrizione di un annuncio, vengono utilizzati conversioni dell’input, con funzioni mysql_real_escape().

4. Javascript-HTML XSS test

JavaScript Injection consiste nell' inserimento dei codici javascript nel form del sistema e una
successiva esecuzione al momento della visualizzazione.
Facciamo un esempio: un utente inserisce il codice javascript come commento ad un annuncio (è possibile inserire qualsiasi tipo di carattere). L'utente che ha inserito l'annuncio, una volta aperta la pagina per visualizzare il commento, involontariamente esegue il codice javaScript e viene reindirizzato alla pagina di login. Altro tipo di attacco potrebbe essere nell'utilizzo di XSS, ad esempio un codice inserito nella form come commento provocherà lo stesso effetto del js sopra descritto. Evocinema prevede questi tipi di attacco, quindi qualsiasi input nel sistema dove sono necessari tutti tipi di carattere (commenti o descrizioni) vengono ripuliti dal codice HTML con funzione testInput().

5. Privilege Escalation test

Il Privilege Escalation consiste nel tentativo di ottenere i privilegi più alti nel sistema. Ad esempio, un utente potrebbe tentare di eseguire una richiesta alle pagine del moderatore o amministratore. Evocinema ha adattato il sistema del routing. Qualsiasi richiesta al sistema, viene reindirizzata al router (index.php). Però prima che l'utente venga reindirizzato alla pagina viene eseguita una funzione, checkPermission(). Questa funzione controlla se un utente può o meno accedere ad una determinata pagina. In caso che l'utente non può accedere a quella pagina viene immediatamente reindirizzato alla home di Evocinema.

6. Recovery testing

La consistenza del sistema in generale è garantita dal fatto che qualsiasi dato persistente viene salvato nel DB, ed ogni operazione è atomica. Quindi nel caso di fallimento, all'utente sarà visualizzato il messaggio di fallimento della richiesta e potrà riprovare.

7. Conclusione

Durante lo sviluppo di Evocinema sono state adottate diverse tecniche per garantire la sicurezza e stabilità del sistema stesso. Tutte le tecniche citate in questo documento sono state testate e all'atto del rilascio del sistema tutto risulta funzionante e coerente con i requisiti non funzionali definiti all'interno del requirements analysis document.