**Università di Salerno**

**Corso di Ingegneria del Software**

**ClickEat**

**Security and Recovery Testing**

**Versione 1.0**



Data

|  |  |
| --- | --- |
| Progetto: ClickEat | Versione: 1.0 |
| Documento: Security and Recovery Testing | Data: 09/02/2019 |

**Partecipanti:**

|  |  |
| --- | --- |
| Nome | Matricola |
| Cupito Andrea [CA] | 0512104538 |
| Amoriello Luca [AL] | 0512104658 |
| Pasquariello Giovanni [PG] | 0512105020 |
| Russo Vincenzo [RV] | 0512104130 |

|  |  |
| --- | --- |
| Scritto da: | Cupito Andrea |

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| Data | Versione | Descrizione | Autore |
| 09/02/2019 | 1.0 | Stesura del documento di Security and Recovery Testing | Membri del Team |

Sommario

Nessuna voce di sommario trovata.

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 ClickEat, 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.

ClickEat interagisce con ‘utente, che può inserire dei dati, e quindi potenzialmente effettuare

una SQLi. La SQLi in sé, 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 ClickEat, 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 <img src=”/logout”/> provocherà lo stesso effetto del js sopra

descritto. ClickEat 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.

ClickEat ha adattato il sistema del routing. Qualsiasi richiesta al sistema, viene reindirizzata al router. 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 ClickEat.

**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 ClickEat 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.