**Eötvös Loránd Tudományegyetem**

Informatikai Kar

*Programozási nyelvek és Fordítóprogramok Tanszék*

Nem biztonságos kód biztonságossá transzformálása a RefactorErl

segítségével

**Témavezetők:** **Szerző:**

Dr. Tóth Melinda, Dr. Bozó István Juhász Alexandra Nelli

Egyetemi docens, Tudományos munkatárs Programtervező informatikus MSc.

**Budapest, 2025**

Témabejelentő / Absract (azt újra kell írni)

A dolgozat célja olyan automatikus, illetve részben automatikus transzformációk definiálása a RefactorErl statikus elemző eszközben, amelyek az Erlang nyelvben előforduló sérülékeny kódrészleteket biztonságossá alakítják. Az Erlang funkcionális, magas szintű programozási nyelv, amelyet gyakran használnak kritikus rendszerek – például banki, telekommunikációs és közlekedés szoftverek – fejlesztésére, így különösen fontos, hogy a forráskódokban ne maradjanak kiaknázható sérülékenységek. Statikus elemzési módszerekkel az ilyen pontok felderíthetők.

A RefactorErl egy nyílt forráskódú statikus elemző eszköz, amely támogatja az Erlang fejlesztők munkáját különféle kódelemzési és keresési funkciókkal, valamint lehetőséget biztosít refaktorálások és transzformációk megvalósítására is. A kutatás fő része azoknak a biztonságossá tevő mintáknak a meghatározása és implementálása, amelyek a program eredeti funkcionalitását megőrizve kizárják a potenciális sérülékenységeket. Ilyen például az inputok megfelelő ellenőrzése beszúrásos támadások megelőzésére.

Az eredményként létrehozott transzformációk célja a fejlesztési folyamat biztonságának növelése és az emberi hibákból fakadó kockázatok csökkentése, ezzel hozzájárulva a megbízhatóbb Erlang alapú rendszerek megvalósításához.

Tartalomjegyzék

[1. Bevezetés (3-5 oldal) 4](#_Toc208706091)

[1.1. A kutatás jelentősége 4](#_Toc208706092)

[1.2 Célkitűzések 4](#_Toc208706093)

[2. Irodalmi áttekintés (10-12 oldal) 5](#_Toc208706094)

[2.1. Forráskód biztonsága 5](#_Toc208706095)

[2.2. Statikus analízis szerepe a kódbiztonságban 5](#_Toc208706096)

[2.3. Erlang biztonsági szempontból 5](#_Toc208706097)

[2.4. RefactorErl további használati területei 5](#_Toc208706098)

[3. Sérülékeny kódrészletek és biztonsági minták (8-10 oldal) 6](#_Toc208706099)

[3.1. Gyakori biztonsági problémák az Erlangban 6](#_Toc208706100)

[3.2. Jellemző sérülékeny kódrészletek 6](#_Toc208706101)

[3.3. Biztonságossá tevő minták 6](#_Toc208706102)

[4. Sérülékeny Erlang kódok átalakításának módszertana RefactorErl segítségével (10-12 oldal) 7](#_Toc208706103)

[4.1. RefactorErl működése és komponensei 7](#_Toc208706104)

[4.2. A transzformációk definiálásának lehetőségei 7](#_Toc208706105)

[5. Megvalósítás és eredmények (10-12 oldal) 8](#_Toc208706106)

[5.1. Implementált transzformációk bemutatása 8](#_Toc208706107)

[5.2. A transzformációk tesztelése 8](#_Toc208706108)

[6. Összegzés és továbbfejlesztési lehetőségek (3-4 oldal) 9](#_Toc208706109)

[Irodalomjegyzék 10](#_Toc208706110)

[Forráskód jegyzék 11](#_Toc208706111)

# 1. Bevezetés (3-5 oldal)

## 1.1. A kutatás jelentősége

## 1.2 Célkitűzések

# 2. Irodalmi áttekintés (10-12 oldal)

## 2.1. Forráskód biztonsága

## 2.2. Statikus analízis szerepe a kódbiztonságban

## 2.3. Erlang biztonsági szempontból

## 2.4. RefactorErl további használati területei

# 3. Sérülékeny kódrészletek és biztonsági minták (8-10 oldal)

## 3.1. Gyakori biztonsági problémák az Erlangban

## 3.2. Jellemző sérülékeny kódrészletek

## 3.3. Biztonságossá tevő minták

# 4. Sérülékeny Erlang kódok átalakításának módszertana RefactorErl segítségével (10-12 oldal)

## 4.1. RefactorErl működése és komponensei

## 4.2. A transzformációk definiálásának lehetőségei

4.3. Általam alkalmazott módszerek

# 5. Megvalósítás és eredmények (10-12 oldal)

## 5.1. Implementált transzformációk bemutatása

## 5.2. A transzformációk tesztelése

5.3. Eredmények értékelése

# 6. Összegzés és továbbfejlesztési lehetőségek (3-4 oldal)

# Irodalomjegyzék

A könyvek idézésének formája a következő:  
[<hivatkozási azonosítója>] <szerző neve>: <a könyv címe>, <a kiadó neve>, <a kiadás éve>, [<terjedelme>], <ISBN szám>. Például: [2] Jan Sommerville: Software Engineering, Addison Wesley, 1988, [742], ISBN-0201- 42765-6.  
Folyóiratcikk idézésének formája a következő:  
[<hivatkozási azonosítója>] <szerző neve>: <a cikk címe>, <a folyóirat címe>, <Vol. száma>, <évszám>, [terjedelme].  
Hálózaton elérhető forrásmunka esetén a „link” megadása után adjuk meg az **elérés dátumát**!

# Forráskód jegyzék