

DIPLOMAMUNKA TÉMABEJELENTŐ

Hallgató adatai:

Név: Juhász Alexandra Nelli

Neptun kód: FHHA98

Képzési adatok:

Szak: programtervező informatikus, mesterképzés (MA/MSc)

Tagozat : Nappali

Belső témavezetővel rendelkezem

Témavezető neve: Dr. Tóth Melinda / Dr. Bozó István

munkahelyének neve, tanszéke: ELTE IK, Programozási nyelvek és Fordítóprogramok Tanszék

munkahelyének címe: 1117, Budapest, Pázmány Péter sétány 1/C.

beosztás és iskolai végzettsége: Egyetemi docens / Tudományos munkatárs, Informatikatudományi PhD / Informatikatudományi PhD

A diplomamunka címe: Nem biztonságos kód biztonságossá transzformálása a RefactorErl segítségével

A diplomamunka témája:

(A témavezetővel konzultálva adja meg 1/2 - 1 oldal terjedelemben diplomamunka témájának leírását)

A diplomamunkám és a kutatásom célja olyan automatikus/részben automatikus transzformációk definiálása a RefactorErlben, melyek a nem biztonságos vagy sérülékeny kódrészleteket biztonságossá alakítják az Erlang programozási nyelvben.

Az Erlang egy elosztott, skálázható, hibatűrő programok fejlesztését biztosító programozási nyelv. Funkcionális jellegű, magasszintű nyelv, melyben ugyan nem lehetetlen, de kevés lehetőség van arra, hogy biztonsági sérülékenységeket hagyjanak a programozók. Azonban a nyelvet gyakran használják kritikus infrastruktúrák (banki rendszerek, telekommunikáció, közlekedés irányító stb.) szoftvereinek megvalósításához, így különösen fontos az, hogy a forráskódokban ne maradjanak kiaknázható támadási pontok, melyek komoly biztonsági kockázatot jelenthetnek. Statikus elemzési módszerekkel az ilyen pontok felderíthetők.

A RefactorErl egy nyílt forráskódú statikus elemző eszköz, amely az Erlang fejlesztők mindennapi munkáját támogatja különféle kódelemzési és keresési funkciókkal. A kutatásom szempontjából azonban a legfontosabb, hogy képes azonosítani a biztonsági kockázatot jelentő kódrészleteket és lehetőséget biztosít forráskód transzformációk/refaktorálások definiálására is.

A RefactorErl által feltárt, Erlangban előforduló sérülékeny kódrészletek egy része olyan, mely a funkcionalitás megtartása mellett átalakítható lenne úgy, hogy a sérülékenység ne legyen kiaknázható. Például, egy beszűrős támadás esetén egy megfelelő input ellenőrzéssel biztonságossá tehető az adott kód. Célom, hogy az ilyen biztonságossá tevő mintákat definiáljam és transzformációkat adjak a RefactorErlben, melyek a nem biztonságos kódrészletek biztonságossá alakítják. Ezek az automatikus transzformációk minimalizálják az emberi beavatkozást, hozzájárulva a hatékonyabb és biztonságosabb fejlesztési folyamatokhoz.

Budapest, 2024. 10. 13.