Erlang programok statikus elemzése, Secure Coding Erlangban

RefactorErl Kutatócsoport

Eötvös Loránd Tudományegyetem, Informatikai Kar

2020. november 4.

National Research, Development and Innovation Office
Hungary

PROGRAM
FINANCED FROM
THE NRDI FUND

A mai napon...

RefactorErl

Secure Coding

Secure Coding támogatása RefactorErl segítségével

Tervek

National Research, Development and Innovation Office
Hungary

PROGRAM
FINANCED FROM
THE NRDI FUND

Erlang, RefactorErl

- Funkcionális, elosztott, konkurens
- Skálázható, magas hibatűrésű
- Számos területen alkalmazzák

- Statikus forráskódelemzés és transzformálás
- Kódmegértés, fejlesztés, karbantartás támogatás
- Szemantikus lekérdező nyelv, Függőségi elemzések, Adatfolyam elemzés, Metrikák, stb.

National Research, Development and Innovation Office
Hungary
PROGRAM FINANCED FROM THE NRDI FUND

Kutatási alprojektek, 2020 (1)

- RefactorErl ipari használhatóságának elősegítése
  - Design rule-ok megfogalmazása
  - Új lekérdezések
  - Hibajavítások, fejlesztések
- **Secure coding**
- Elosztott programok elemzése és refaktorálása
- Duplikált kódok keresése
- Gráfok interaktiv vizualizációja a kódmegértés támogatására

National Research, Development and Innovation Office
Hungary
PROGRAM
FINANCED FROM
THE NRDI FUND

Kutatási alprojektek, 2020 (2)

- Rekurzió hatása a programok bonyolultságára
- Forráskódok változásának nyomkövetése az SPG-ben
- Stílusátalakítás refaktorálással az optimalizáció jegyében (hatékonyság, olvashatóság, komplexitás)
- Elixir programok elemzése
- Adatbázisok hatékonyságának vizsgálata
- Erlang fejlesztők programozói stílusjegyeinek a detektálása.

National Research, Development and Innovation Office
Hungary
PROGRAM FINANCED FROM THE NRDI FUND

Secure Coding

- Szoftverek sérülékenysége: nem szándékos programozó hiba?
- Imperatív nyelvek esetén számos eszköz áll a fejlesztők rendelkezésére a sebezhetőségek kiszűrésére (CodeChecker, SpotBugs, SonarQube, Fortify)
- SEI-CERT Secure Coding Standards
- OWASP (Open Web Applications Security Project) rules
- Deklaratív nyelvek (Erlang) esetén kevésbé elterjedt
- Egyre nagyobb az igény a szoftveres megoldásokra

```
-module(injection).
-export([run_cmd/1]).

run_cmd(Input) ->
    os:cmd("cat "++ Input).
```

→

```
1>injection:run_cmd("test.txt;ls").
"Hello World! test.txt"
```

NATIONAL RESEARCH, DEVELOPMENT AND INNOVATION OFFICE HUNGARY
PROGRAM FINANCED FROM THE NRDI FUND

Hogyan?

- Futási idejű monitorozás
- Manuális ellenőrzés (code review, security testing)
- Szigorított építkezés (nyelvi vagy fordítóprogram támogatása)
- Statikus vagy dinamikus ellenőrző eszközökkel

```
-module(injection).
-export([run_cmd/1]).

run_cmd(Input) ->
    os:cmd("cat "++ Input).
```

```
1>injection:run_cmd("test.txt;ls").
"Hello World! test.txt"
```

NATIONAL RESEARCH, DEVELOPMENT AND INNOVATION OFFICE HUNGARY
PROGRAM FINANCED FROM THE NRDI FUND

Sérülékenységek Erlangban

- Nincs adatbázis a sérülékenységek listázására
- A cégek saját fejlesztőiket bízzák meg ellenőrzésekkel (security expert)
- 2020 februárjától EEF Security Working Group (Erlang Ecosystem Foundation)
- PEST (Primitive Erlang Security Tool) alap ellenőrzései (string matching)
- Cél: RefactorErl mély szemantikus elemzéseire építve adjunk eszközt

National Research, Development and Innovation Office
Hungary
PROGRAM FINANCED FROM THE NRDI FUND

Erlang sérülékenységek csoportosítása

- Együttműködési képességből fakadó sebezhetőségek
- Konkurens programozásból eredő hibalehetőségek
- Elosztott programozásból adódó sérülékenységek
- Injection támadási típusok
- Memória túlterhelést eredményező támadási formák

National Research, Development and Innovation Office
Hungary
PROGRAM FINANCED FROM THE NRDI FUND

Együttműködési képességből fakadó sebezhetőségek

- Portokon keresztül:

```
-module(complex1).
-export([start/1, init/1]).

start(ExtPrg) ->
    spawn(?MODULE, init, [ExtPrg]).

init(ExtPrg) ->
    register(complex, self()),
    process_flag(trap_exit, true),
    loop(open_port({spawn, ExtPrg}),
         [{packet, 2}])).
```

- Dinamikusan betölthető könyvtárak segítségével:

```
-module(complex2).
-export([foo/1]).
-on_load(init/0).

init() ->
    {ok, ExtPrg} = io:read("Provide a program..."),
    ok = erlang:load_nif(ExtPrg, 0).

foo(_X) -> exit(nif_library_not_loaded).
```

National Research, Development and Innovation Office Hungary
PROGRAM FINANCED FROM THE NRDI FUND

Konkurens programozásból eredő hibalehetőségek

- Folyamatok indításának és összekapcsolásának két műveletben történő végrehajtása

```
chain(0) ->
    receive
        _ -> ok
    after 2000 ->
        exit("chain dies here")
    end;
chain(N) ->
    Pid = spawn(fun()->chain(N-1) end),
    link(Pid),
    receive
        _ -> ok
    end.
```

- Folyamatok prioritásának módosítása:

```
process_flag(priority, high).
```

- ETS táblák bejárása a tábla rögzítése nélkül:

```
ets:first/1, ets:next/2, ets:last/1,
ets:select/1-3, ets:select_count/2, ...
```

National Research, Development and Innovation Office Hungary
PROGRAM FINANCED FROM THE NRDI FUND

Elosztott programozásból adódó sérülékenységek

- Hálózati rendszermaggal kapcsolatos függvényhívások:

```
net_kernel:allow/1,
net_kernel:connect_node/1,
net_kernel:start/1
```

- SSL-3.0 és TLS-1.0 kommunikációs protokollokra vonatkozó nem megfelelő beállítások használata:

```
ssl:connect("example.net", 443, [
    {padding_check, false},
    {beast_mitigation, disabled},
    {fallback, true}
]).
```

- Elavultnak számító kriptográfiai függvények alkalmazása:

```
crypto:block_encrypt/3/4,
crypto:block_decrypt/3/4,
crypto:cmac/3/4, crypto:hmac/3/4, ...
```

National Research, Development and Innovation Office Hungary
PROGRAM FINANCED FROM THE NRDI FUND

Injection támadási típusok

- OS szintű utasítások ismeretlen bemenettel való hívása:

```
-module(injection).
-export([run_cmd/1]).

run_cmd(Input) ->
    os:cmd("cat "++ Input).
```

- Fájlkezeléssel kapcsolatos műveletek ismeretlen bemenettel való hívása:

```
-module(injection).
-export([eval/1]).

eval(Input) -> file:eval(Input).
```

- Ismeretlen helyről jövő programkódok dinamikus lefordítása és betöltése

```
-module(injection).
-export([load/1]).

load(Input) ->
    code:load_file(Input).
```

NATIONAL RESEARCH, DEVELOPMENT AND INNOVATION OFFICE HUNGARY
PROGRAM FINANCED FROM THE NRDI FUND

Memória túlterhelést eredményező támadási formák

- Atomok dinamikus létrehozásával kapcsolatos függvények:

```
parse_uri(Input) ->
    http_uri:parse(Input, [{ipv6_host_with_brackets, true}]).
```

- XML elemzéssel kapcsolatos függvények megfelelő eseménykezelő (`internalEntityDecl`, `externalEntityDecl`) függvényhívás nélkül:

```
parse_xml(Input) ->
    xmerl_sax_parser:stream(Input, []).
```

National Research, Development and Innovation Office Hungary
PROGRAM FINANCED FROM THE NRDI FUND

Sérülékenységek detektálása a RefactorErl segítségével

Algoritmus 1 OS injection kivédésére szolgáló algoritmus

Function get_calls_for_os(Fun)

1: Matchers $\leftarrow [\{os, [cmd, putenv]\}]$
2: OsFuns $\leftarrow get\_calls\_for(Fun, Matchers)$
3:
4: FunParamTuples $\leftarrow new\_list$
5: **for** $O \in OsFuns$ **do**
6: Param $\leftarrow get\_expr\_params\_for(O, Matchers, get\_all\_children(O))$
7: $add\_to\_list(\{O, Param\}, FunParamTuples)$
8: **end for**
9:
10: DataFlowTuples $\leftarrow get\_origins(FunParamTuples)$
11: NonSafeDataFlowTuples $\leftarrow get\_unsafe\_funs(DataFlowTuples)$
12: FunsWithNoBody $\leftarrow get\_funs\_no\_body(NonSafeDataFlowTuples)$
13: ExportedFuns $\leftarrow get\_exported\_funs(NonSafeDataFlowTuples)$
14: **return** $merge(FunsWithNoBody, ExportedFuns)$

PROGRAM
FINANCED FROM
THE NRDI FUND

Néhány részlet

- Szemantikus Programgráf bejárása
- A gyanús függvényhívások azonosítása
- Eredmények szűrése és ellenőrzése legtöbbször adatfolyam elemzéssel
- Részeredmények figyelembe vétele
- Megengedett műveletek megadása

National Research, Development and Innovation Office
Hungary
PROGRAM
FINANCED FROM
THE NRDI FUND

Biztonsági rések feltárása a RefactorErl szemantikus lekérdező nyelvének segítségével

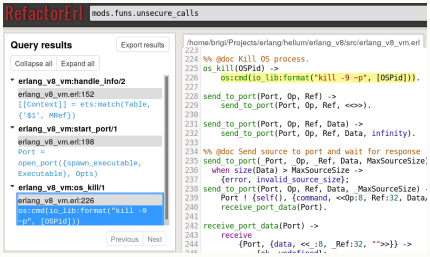
- Szintaktikai és szemantikai információt ad az Erlang programokkal kapcsolatban
- A lekérdező nyelv alkotórészei megfeleltethetőek az Erlang szemantikus nyelvi elemeinek: fájlok, függvények, kifejezések, változók...
- `mods.funs.calls`

National Research, Development and Innovation Office
Hungary
PROGRAM FINANCED FROM THE NRDI FUND

Biztonsági rések feltárása a RefactorErl szemantikus lekérdező nyelvének segítségével

- Bemenet: a függvény típusú entitások
- Kimenet: az érzékeny kifejezések halmaza
- Példa sérülékenységek lekérdezésére:

```
mods[name=foo].funs[name=bar].unstable_call
mods[name=foo].funs[name=bar].unsafe_network
mods[name=foo].funs[name=bar].unsecure_calls
```



NATIONAL RESEARCH, DEVELOPMENT AND INNOVATION OFFICE
HUNGARY
PROGRAM FINANCED FROM THE NRDI FUND

Lekérdezések 1.

Selectors	Short description
unsecure_calls	Lists all the possible vulnerabilities
unsecure_interoperability	Lists interoperability related weaknesses
unsecure_concurrency	Identifies concurrency related issues
unsecure_os_call	Checks for OS injection
unsecure_port_creation	Identifies port creation related issues
unsecure_file_operation	Lists unsecure file handling
unstable_call	Shows possible atom exhaustion
nif_calls	Identifies unsecure NIF calls
unsecure_port_drivers	Lists the unsecure ddll usage

1. táblázat: The available security checks in RefactorErl (1)

NATIONAL RESEARCH, DEVELOPMENT AND INNOVATION OFFICE HUNGARY
PROGRAM FINANCED FROM THE NRDI FUND

Lekérdezések 2.

Selectors	Short description
decommissioned_crypto	Lists the legacy functions from crypto module
unsecure_compile_operations	Shows unsecure compile/code loading related operations
unsecure_process_linkage	Lists unsecure process linkage
unsecure_prioritization	Identifies unsecure process prioritization
unsecure_ets_traversal	Lists unsecure ETS traversal
unsafe_network	Checks for unsecure kernel related operations
unsecure_xml_usage	Identifies unsecure xml parsing
unsecure_communication	Lists unsecure communication related settings

2. táblázat: The available security checks in RefactorErl (2)

National Research, Development and Innovation Office Hungary
PROGRAM FINANCED FROM THE NRDI FUND

Validáció

Szoftver neve	Azonosított hibák száma	Azonosított hibák típusa	Vizsgált sorok száma	A vizsgálat lefutási ideje
helium/erlang_v8	3	*unsecure_ets_traversal (1), unsecure_port_creation (1), unsecure_os_call (1)*	302	1 mp
helium/miner	7	*unsecure_os_calls (3), unstable_call (4)*	6236	15 mp
heroku/logplex	15	*unsecure_ets_traversal (11), unstable_call (4)*	10634	25 mp
k2informatics/ranch	7	*unsecure_ets_calls (7)*	3052	7 mp
relayr/gen_coap	2	*unstable_call (2)*	2076	6 mp

National Research, Development and Innovation Office
Hungary
PROGRAM FINANCED FROM THE NRDI FUND

Validáció

Pest elemző eredménye a `relayr/gen_coap` projekten futtatva:

```
brigi@debVM:~/Projects/erlang$ ~/Projects/pest/pest/pest.erl -r -s 0 relayr/gen_coap/_build/default/lib/gen_coap/ebin/
 15: Keep OpenSSL updated for crypto module use (run with "-V crypto")
     coap_dtls_listen.beam:19 (ssl:_/_)
     coap_dtls_socket.beam:[32,43,47,60,64] (ssl:_/_)
brigi@debVM:~/Projects/erlang$
```

RefactorErl elemző eredménye a `relayr/gen_coap` projekten futtatva:

```
(refactorerl@localhost)31> ri:q("mods.funs.unsecure_calls").
coap_client:resolve_uri/1
    {ok, {Scheme, _UserInfo, Host, PortNo, Path, Query}} =
        http_uri:parse(Uri, [{scheme_defaults, [{coap, ?DEFAULT_COAP_PORT},
 {coaps, ?DEFAULT_COAPS_PORT}]}])
coap_server_content:filter/2
    filter(
        case binary:split(Search, <<$=>>) of
            [Name0, Value0] ->
                Name = list_to_atom(binary_to_list(Name0)),
                Value = wildcard_value(Value0),
                lists:filter(
                    fun (Link) -> match_link(Link, Name, Value) end,
                    Links);
            _Else ->
                Links
        end,
        Query)
ok
(refactorerl@localhost)32>
```

NATIONAL RESEARCH, DEVELOPMENT AND INNOVATION OFFICE HUNGARY
PROGRAM FINANCED FROM THE NRDI FUND

Továbbfejlesztési lehetőségek

- További hibatípusok keresése
- Személyreszabhatóság növelése:
  - Biztonsági szint beállításának lehetősége
  - Kivételek megadásának fejlesztése
  - Alkalmazásspecifikus tudás beépítése
- Fals pozitív esetek vizsgálata
- "Erlang CodeChecker"
- CI integráció

National Research, Development and Innovation Office
Hungary
PROGRAM
FINANCED FROM
THE NRDI FUND

Hová haladunk?

- Erlang CodeChecker
  - Design rules, biztonsági és egyéb ellenőrzések
- Mintafelismerés
- Duplikált kódok elemzése
- Ipari alkalmazhatóság növelése
- Hibák forrásának detektálása szimbolikus végerhajtás segítségével
- Green Computing
- Hatáselemzés
- Decompiling
- ...

National Research, Development and Innovation Office
Hungary
PROGRAM FINANCED FROM THE NRDI FUND

Q&A

NATIONAL RESEARCH, DEVELOPMENT AND INNOVATION OFFICE
HUNGARY

PROGRAM FINANCED FROM THE NRDI FUND