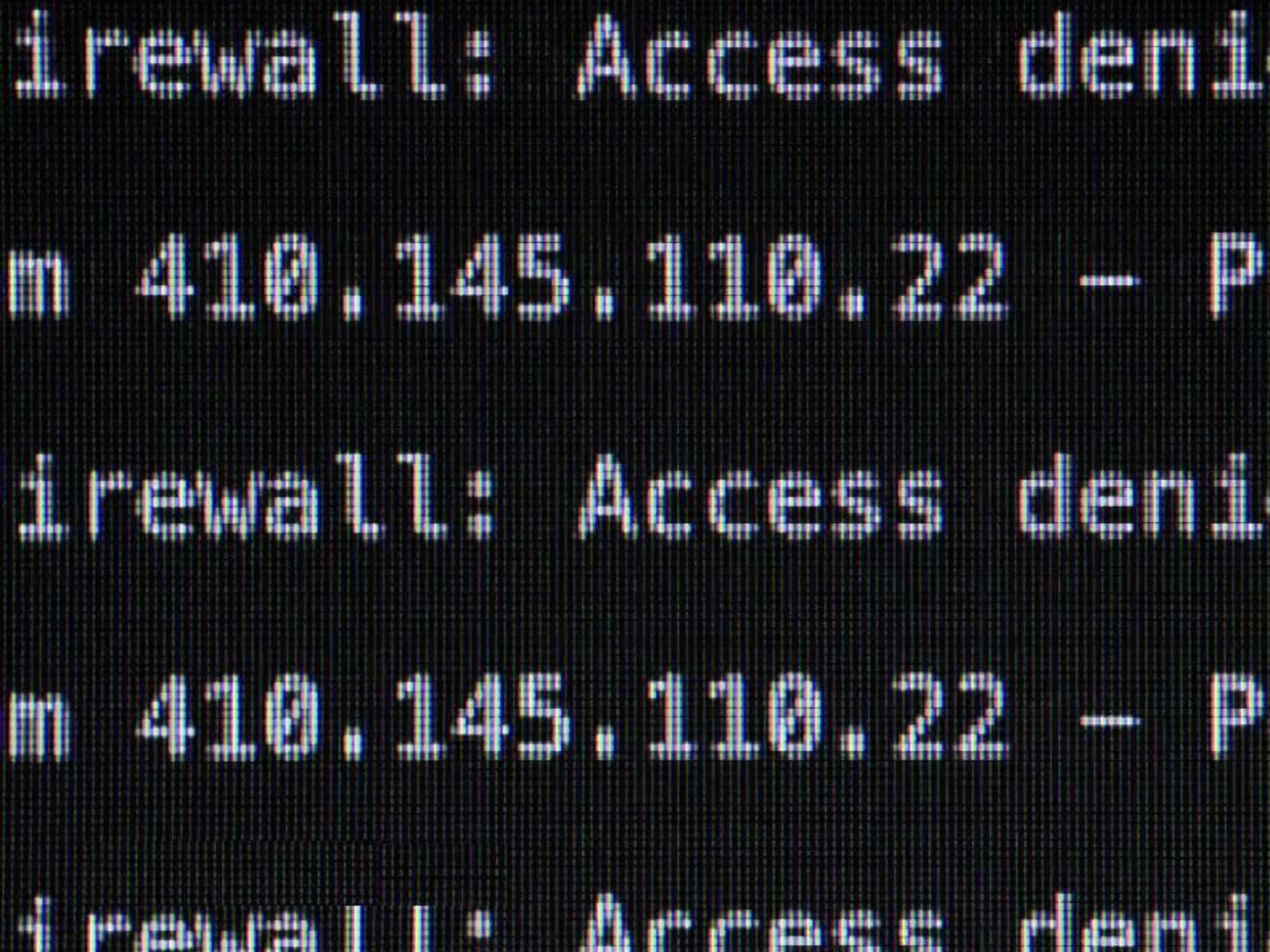
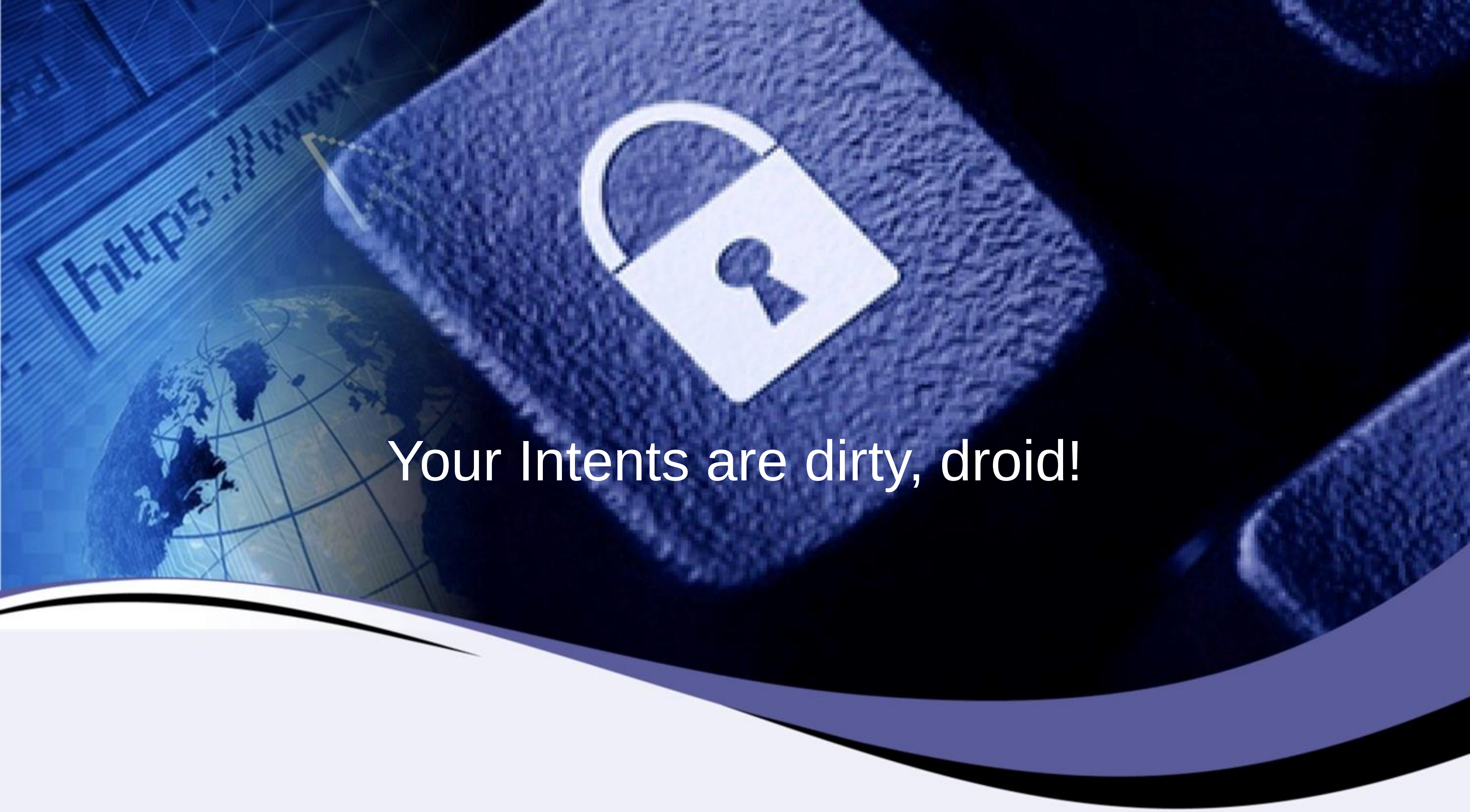


```
irewall: Access deni
m 410.145.110.22 - P
irewall: Access deni
m 410.145.110.22 - P
irewall: Access deni
```



Your Intents are dirty, droid!

OWASP
The Open Web Application Security Project

Razvan Ionescu
razvan.ionescu@intel.com

Cristina Stefania Popescu
cristina.popescu@intel.com

Who are the speakers ?

OWASP
The Open Web Application Security Project

Răzvan
Security QA Engineer @Intel
geocacher, trekker, squash player
Presenter

Ştefania
Security QA Intern @Intel
open-minded, optimistic, resourceful
Demo goddess

Agenda

OWASP

The Open Web Application Security Project

- Motivation
- Existing solution(s)
- intents.fuzzinozer – Intent fuzzing module for Drozer
- SHOW time

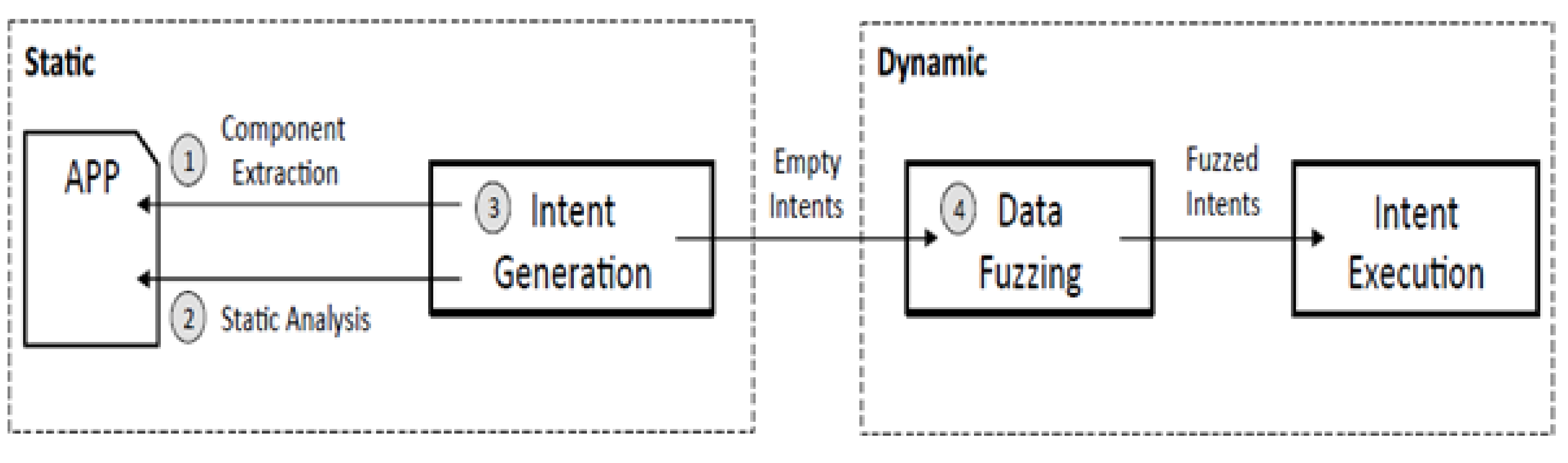
Motivation

TARGET
ACQUIRED!

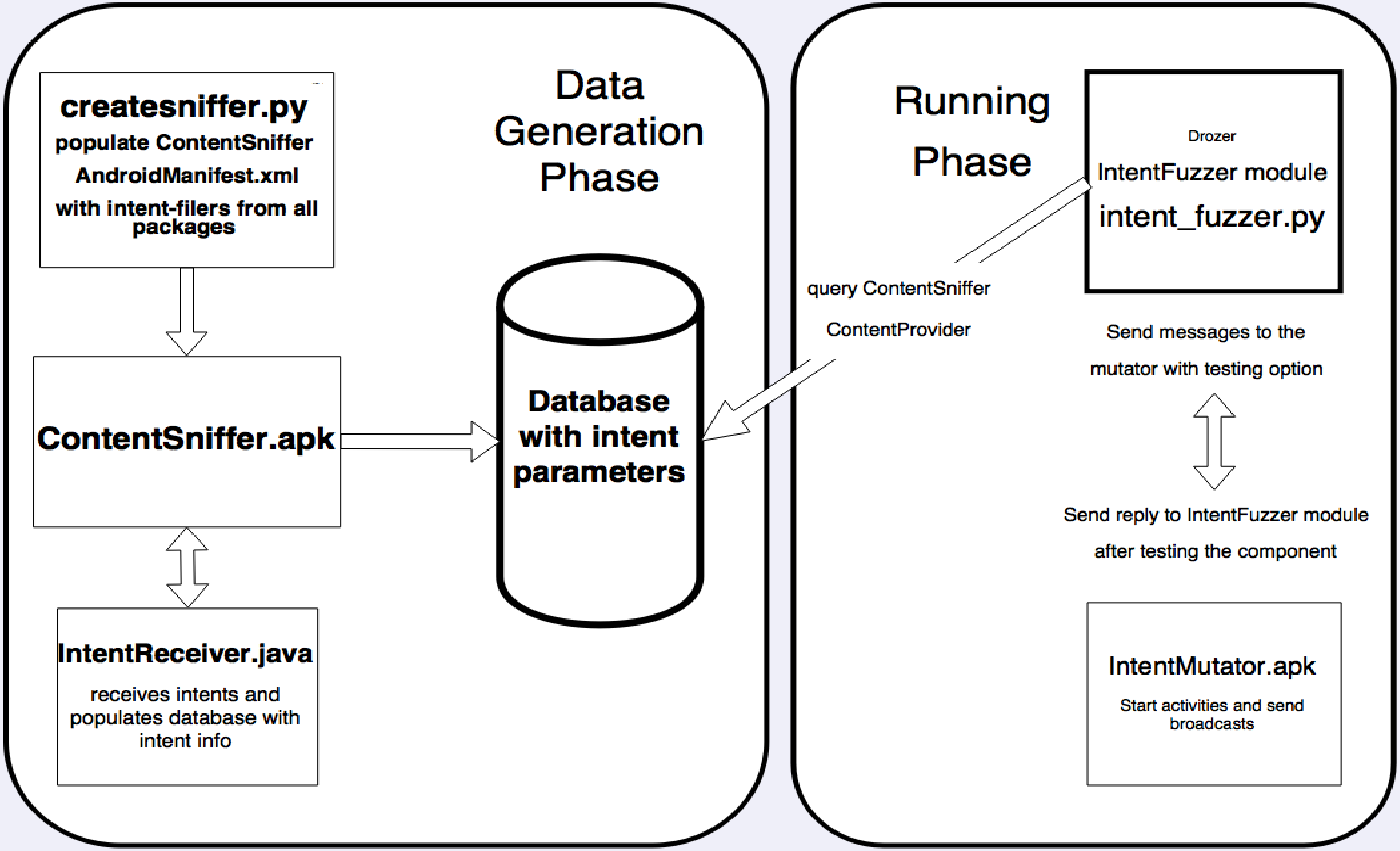
TARGET
ACQUIRED!

Intent? Fuzzing?

```
Intent intent = new Intent(Intent.ACTION_SEND);
intent.setType(“text/plain”);
intent.putExtra(android.content.Intent.EXTRA_TEXT, “Hello!”);
startActivity(intent);
```



Existing tools - intent_fuzzer module arhitecture



Data Generation Phase

createsniffer.py
populate ContentSniffer AndroidManifest.xml with intent-filers from all packages

ContentSniffer.apk

IntentReceiver.java
receives intents and populates database with intent info

Database with intent parameters

Running Phase

Drozer
IntentFuzzer module
intent_fuzzer.py

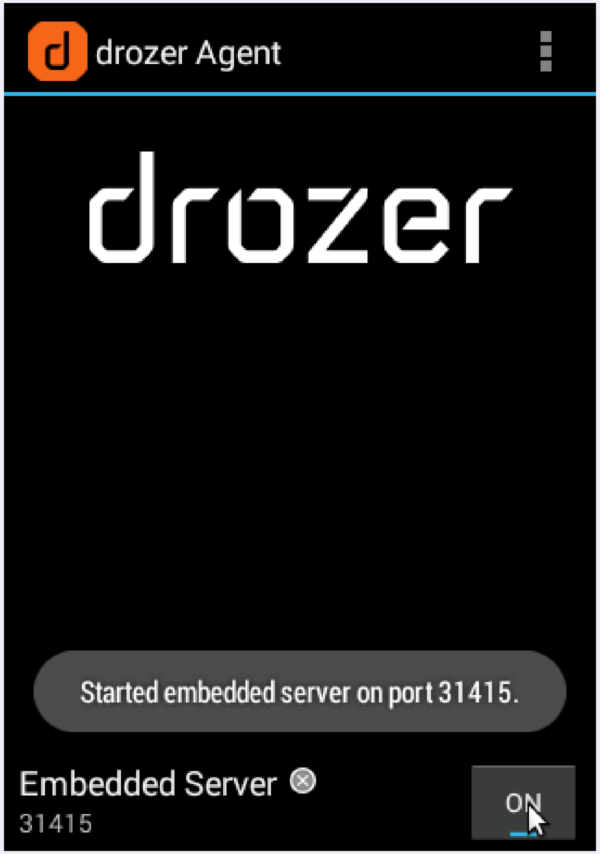
query ContentSniffer ContentProvider

Send messages to the mutator with testing option

Send reply to IntentFuzzer module after testing the component

IntentMutator.apk
Start activities and send broadcasts

Drozer



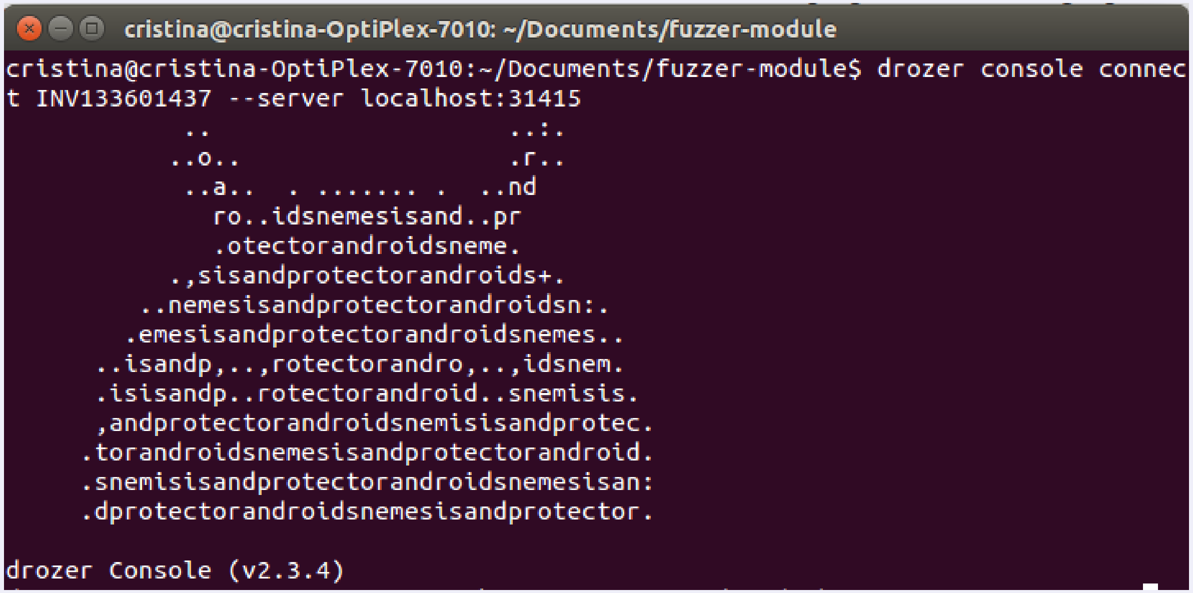
drozer Agent

drozer

Started embedded server on port 31415.

Embedded Server
31415

ON



cristina@cristina-OptiPlex-7010: ~/Documents/fuzzer-module

```
cristina@cristina-OptiPlex-7010:~/Documents/fuzzer-module$ drozer console connec
t INV133601437 --server localhost:31415
            ..                    ..:.
           ..o..                  .r..
            ..a..  . ....... .  ..nd
              ro..idsnemesisand..pr
              .otectorandroidsneme.
           .,sisandprotectorandroids+.
         ..nemesisandprotectorandroidsn:.
        .emesisandprotectorandroidsnemes..
      ..isandp,..,rotectorandro,..,idsnem.
      .isisandp..rotectorandroid..snemisis.
      ,andprotectorandroidsnemisisandprotec.
     .torandroidsnemesisandprotectorandroid.
     .snemisisandprotectorandroidsnemesisan:
     .dprotectorandroidsnemesisandprotector.

drozer Console (v2.3.4)
```

Drozer module

A Drozer module must define the following:

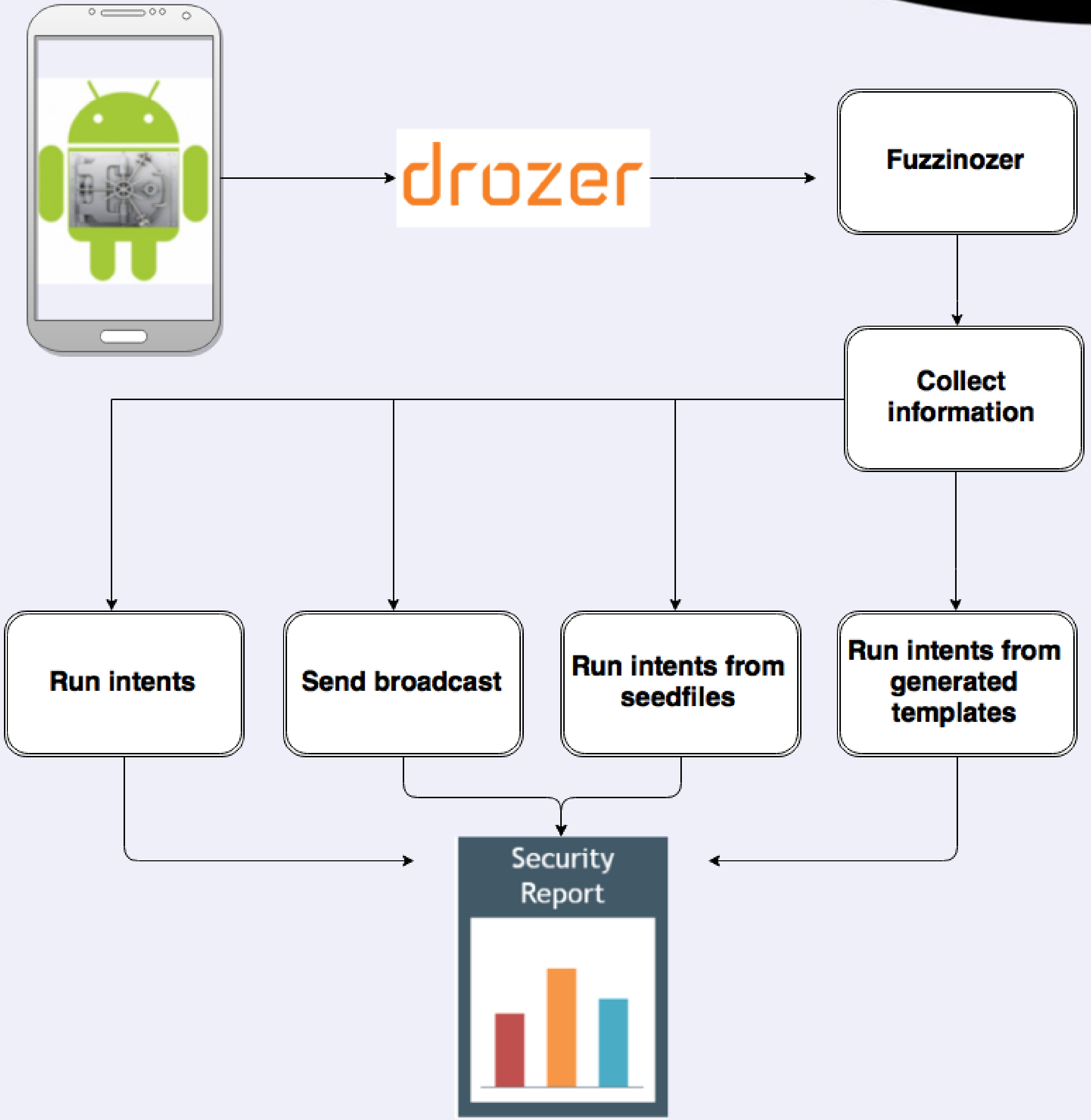
- name (a headline name that describes the module’s purpose)
- description (a longer description of what the module does)
- examples (a few examples of common usage patterns)
- author (the name of the module author, or an array of names)
- date (the date on which the module was last updated)
- license (the license under which this module is released)
- path (an array that describes the namespace of the module)

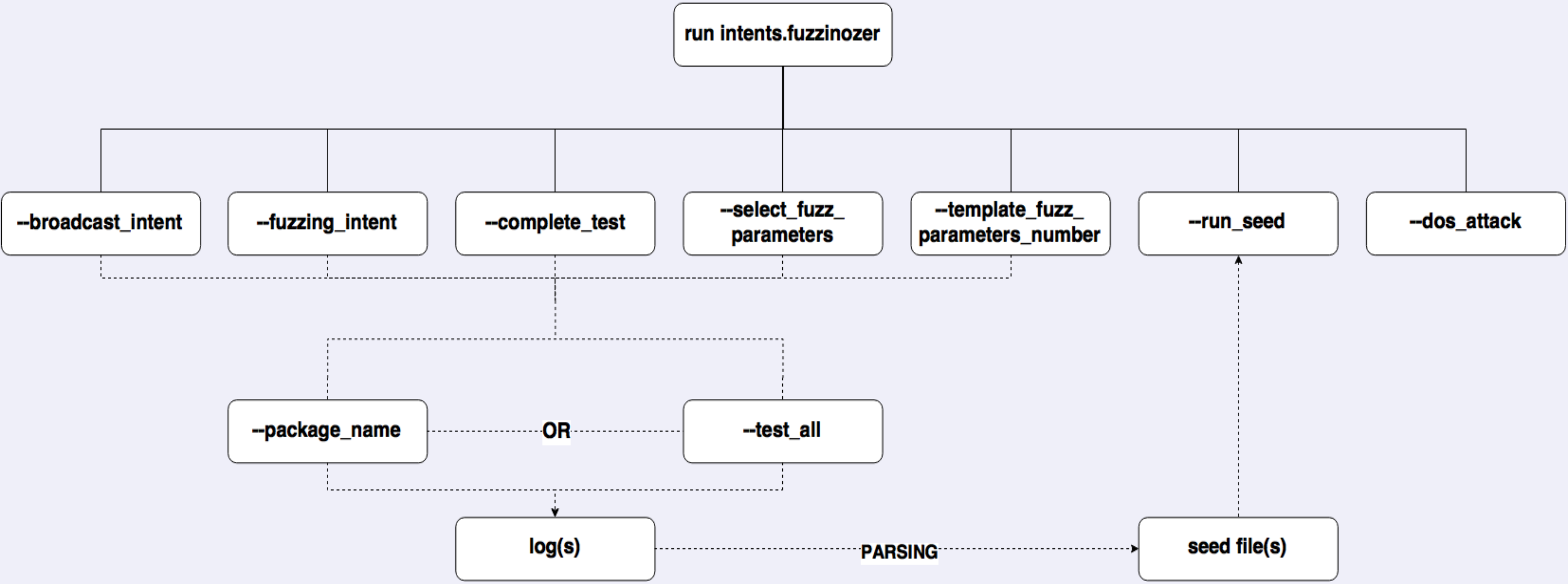
Fuzzinozer

```
class Fuzzinozer(Module,common.PackageManager):
    '''
    Intent_fuzzing module class
    '''
    name = "fuzzinozer"
    description = "Android intent fuzzing module"
    examples = ""
    author = "Popescu Cristina Stefania"
    date = "2015-10-08"
    license = "3 clause BSD"
    path = ["intents"]

    def add_arguments(self, parser):
        parser.add_argument("--package_name", help="specify name of package to test ")
        parser.add_argument("--test_all", action='store_true', help="test all packages")
        parser.add_argument("--broadcast_intent", action='store_true', help="send broadcast ... ")
        parser.add_argument("--fuzzing_intent", action='store_true', help="send intent with ...")
        parser.add_argument("--complete_test", action='store_true', help="test with all ...")
        parser.add_argument("--select_fuzz_parameters", help="give the parameters you want ...")
        parser.add_argument("--run_seed", help="select the seed file you want to run")
        parser.add_argument("--device", help="used only for automated tests")
        parser.add_argument("--template_fuzz_parameters_number", help="give the number of ...")
        parser.add_argument("--dos_attack", help="give the number of intents you want to test")

    def execute(self, arguments):
```

How it works

OWASP
The Open Web Application Security Project



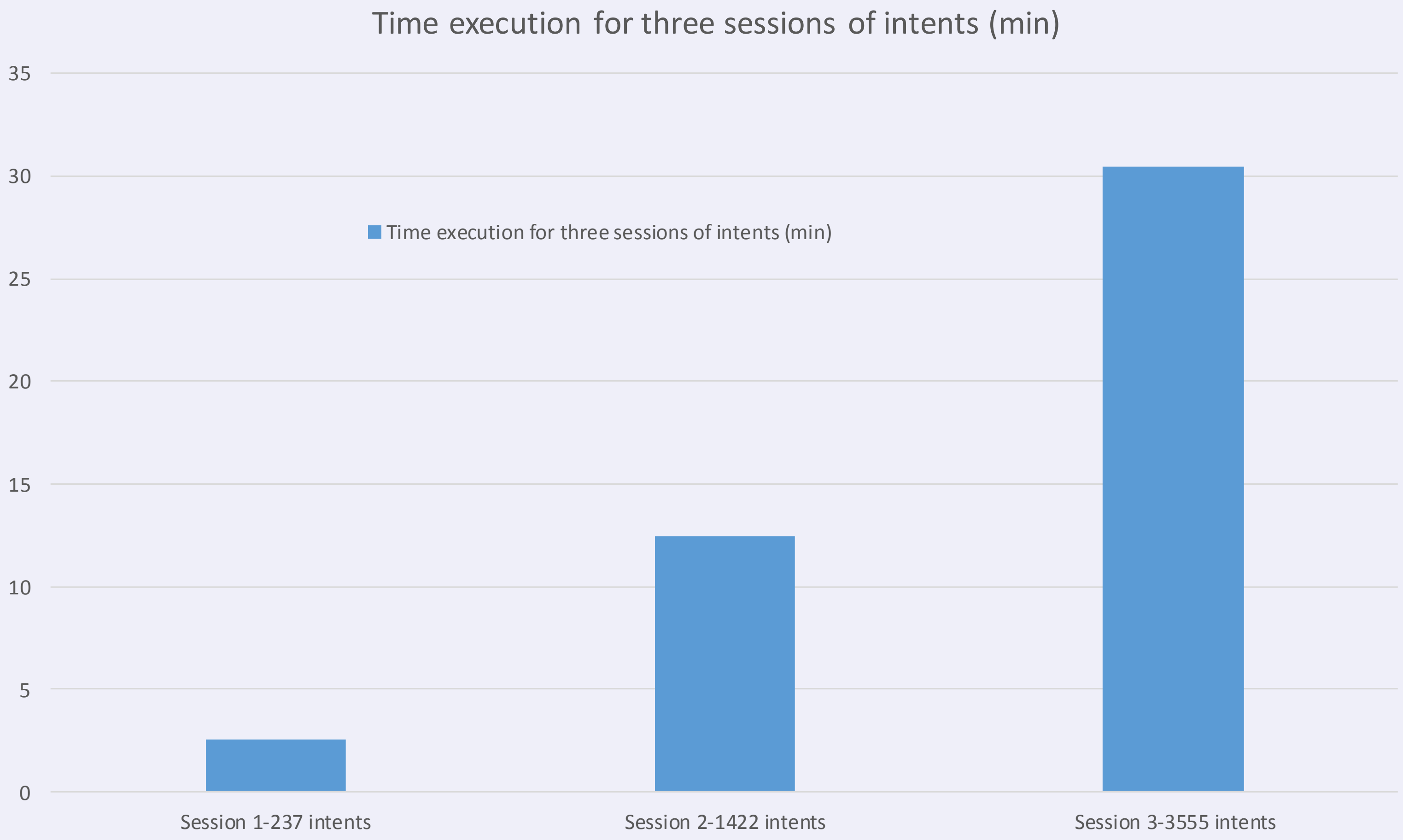

run intents.fuzzinozer
--broadcast_intent
--fuzzing_intent
--complete_test
--select_fuzz_ parameters
--template_fuzz_ parameters_number
--run_seed
--dos_attack
--package_name
OR
--test_all
log(s)
PARSING
seed file(s)

Fuzzinozer running example

```
dz> run intents.fuzzinozer --fuzzing_intent --package_name
com.google.android.gms --template_fuzz_parameters_number 6

dz> run intents.fuzzinozer --complete_test --package_name
com.google.android.gms

dz> run intents.fuzzinozer --run_seed
seedfile_com.google.android.gms_NullPointerException.txt

dz> run intents.fuzzinozer --broadcast_intent --package_name
com.google.android.gms

$ drozer console connect -c “run intents.fuzzinozer --
broadcast_intent --test_all”
```

Results

OWASP
The Open Web Application Security Project



Did it crash?

javaNullPointerException

SecurityException

javaClassNotFoundException

IllegalStateException

DoS attack

ClassCastException

NumberFormatException

ClassCastException

IllegalArgumentException

https://github.com/fuzzing