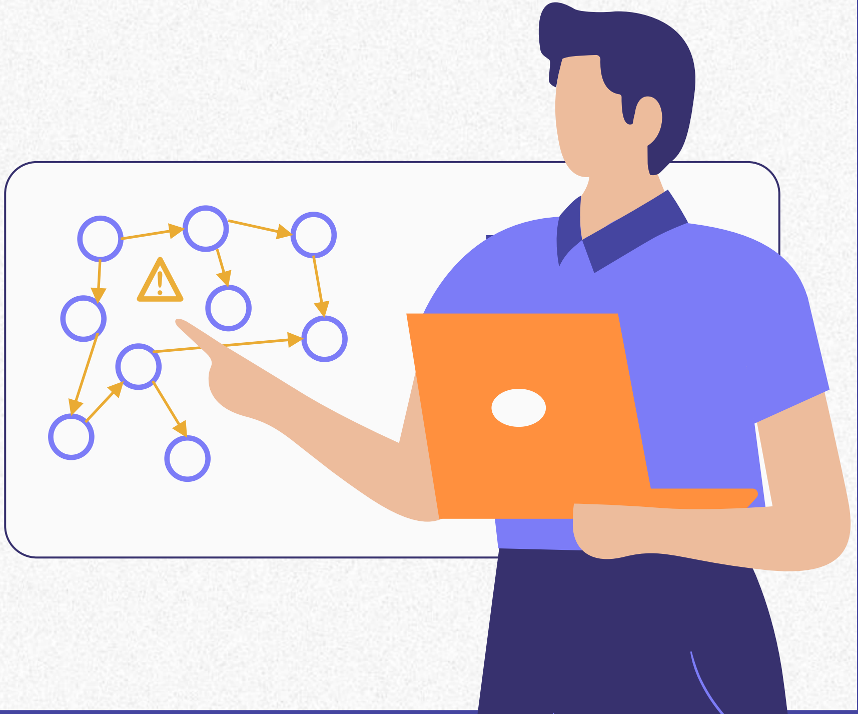
APCD : Vulnerability Detection Using Mixed Graphical Representation



Introduction & Goals

Our approach consists of extracting graphical representations from the source code and then processing all the possible combinations in a Graphical conventional neural network to obtain a multi-vulnerability classification.

Existing Approaches

01 Fuzzing techniques

02 Taint analysis & symbolic executions

03 Source code processing and analysis

04 Natural Language processing

Most vulnerabilities are characterized by more than one aspect and require the code to be treated from more than one angle and with respect to more than one characteristic.

Context & definitions

Context and definitions

AST

Tree representation of the abstract syntactic structure of the code.

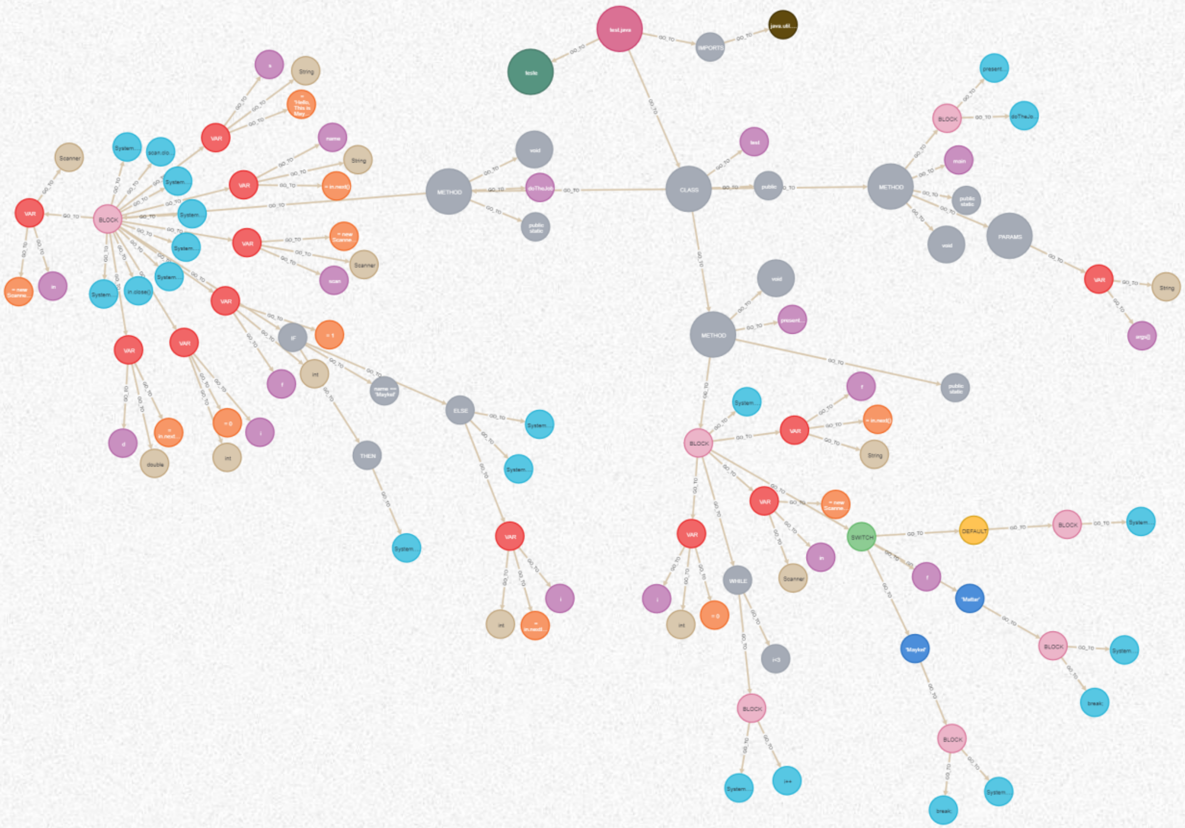
CFG

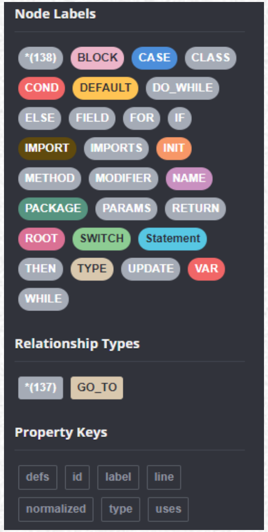
A representation that focuses on the control flow of the system

PDG

Focuses on the data and the control dependence for each operation in a program.

AST





Node Labels

*(138) BLOCK CASE CLASS

COND DEFAULT DO_WHILE

ELSE FIELD FOR IF

IMPORT IMPORTS INIT

METHOD MODIFIER NAME

PACKAGE PARAMS RETURN

ROOT SWITCH Statement

THEN TYPE UPDATE VAR

WHILE

Relationship Types

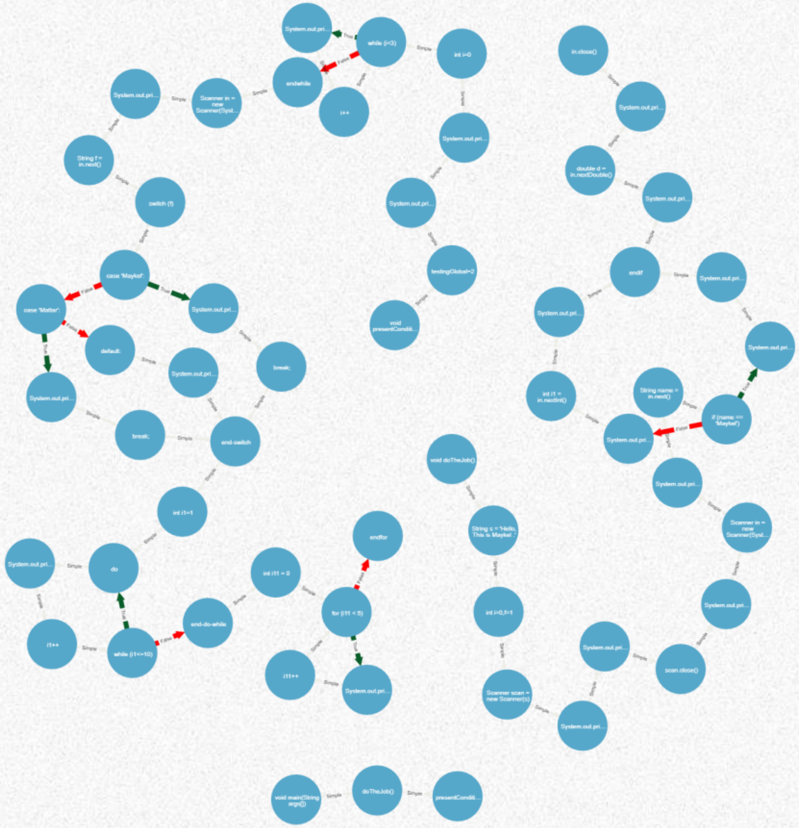
*(137) GO_TO

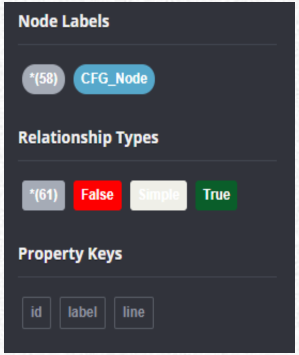
Property Keys

defs id label line

normalized type uses

CFG





Node Labels

*(58) CFG_Node

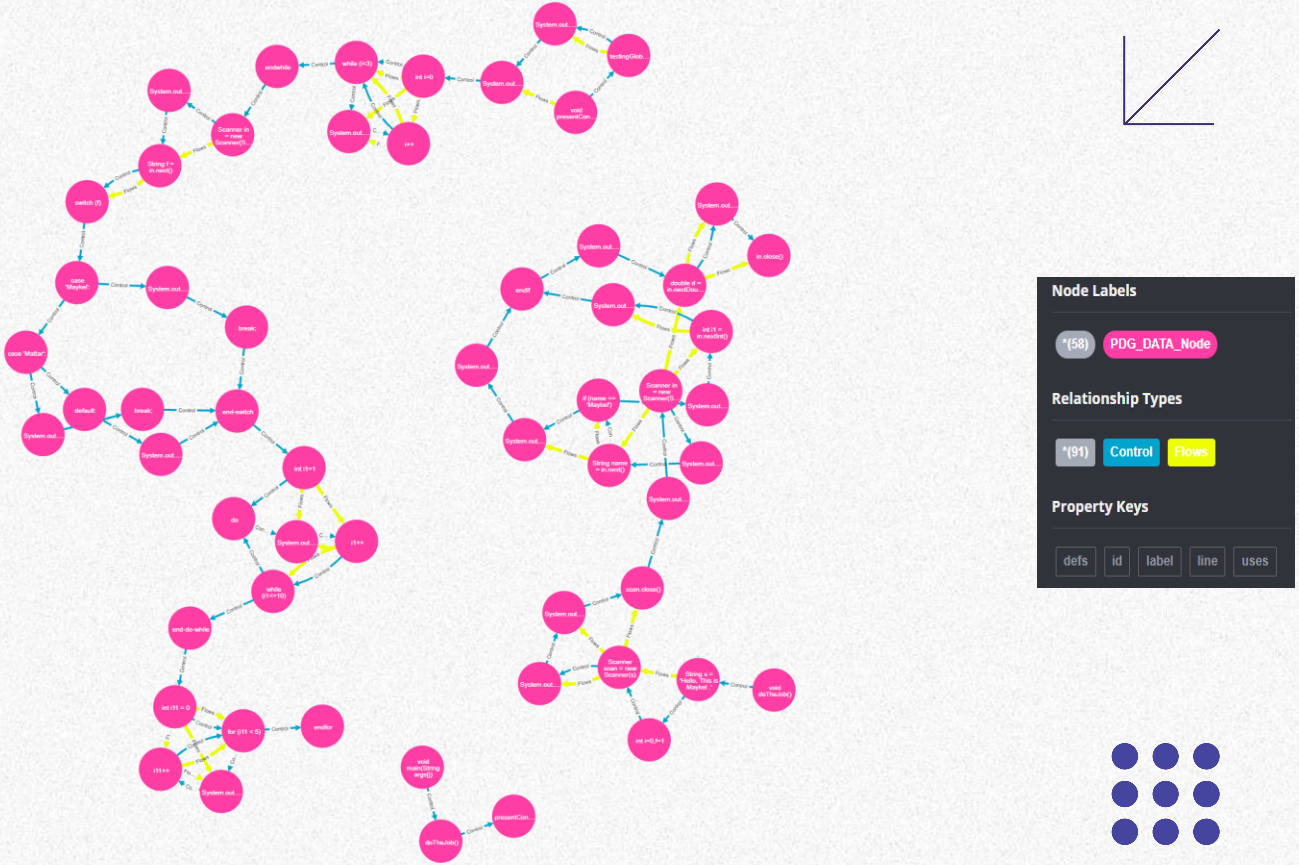
Relationship Types

*(61) False Simple True

Property Keys

id label line

PDG


Node Labels
*(58)
PDG_DATA_Node
Relationship Types
*(91)
Control
Flows
Property Keys
defs
id
label
line
uses
System.out...
testingGlob...
void presentCon...
int i=0
while (i<3)
i++
endwhile
Scanner in = new Scanner(S...
String f = in.next()
switch (f)
case 'Maykel'
break;
case 'Mattar'
default:
end-switch
int i1=1
do
i1++
while (i1<=10)
end-do-while
int i11 = 0
for (i11 < 5)
endfor
i11++
void main(String args[])
doTheJob()
presentCon...
in.close()
double d = in.nextDou...
endif
int i1 = in.nextInt()
if (name == 'Maykel')
String name = in.next()
scan.close()
Scanner scan = new Scanner(s)
String s = "Hello, This is Maykel"
void doTheJob()
int i=0,f=1

Proposal Phase

Phase 1

Extraction

Extract the representations from source code.

Phase 2

Mixing

We clean and combine the representaiton to obtain richer ones

Phase 3

DGCNN

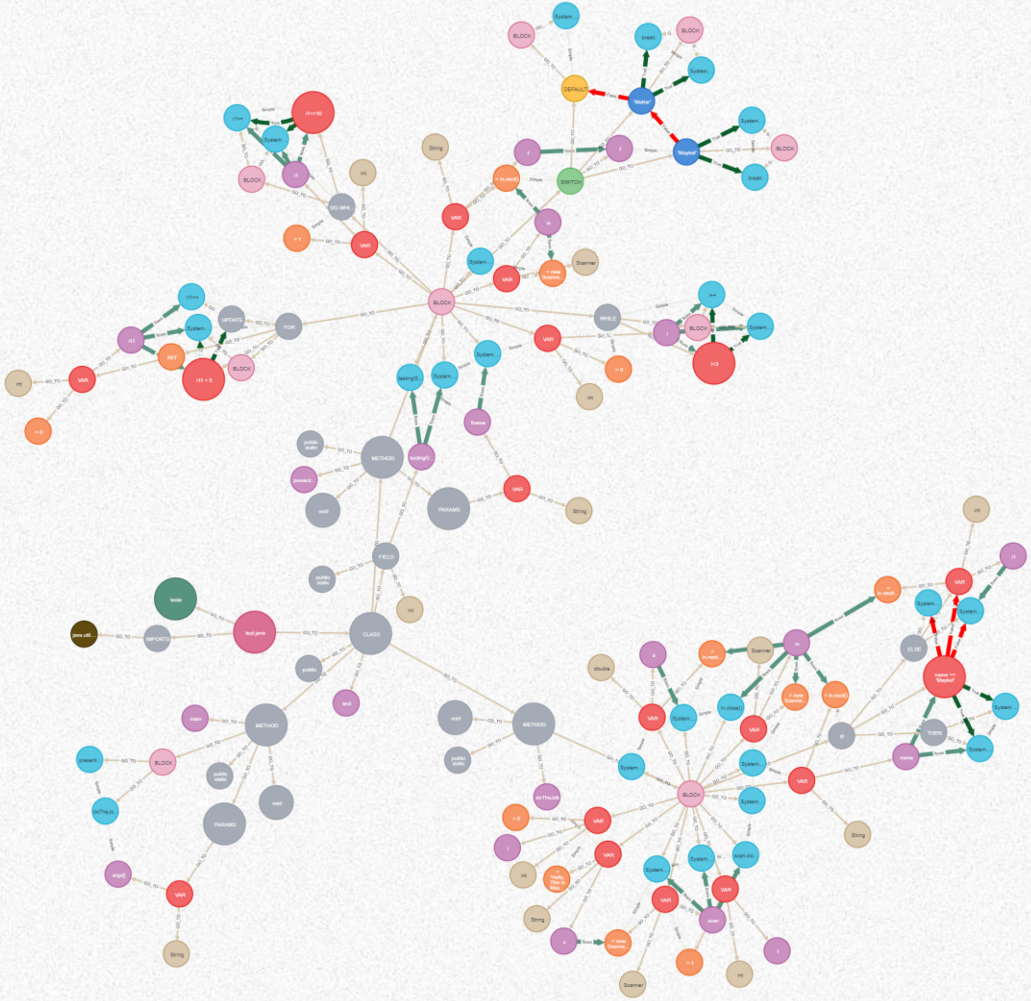
We train a deep graphical conventional neural network.

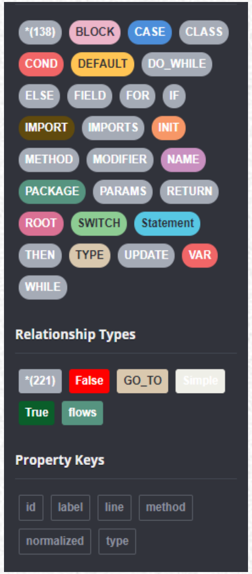
Phase 4

Benchmark

We repeat the previous phase for all the combination and benchamrk it.

APC





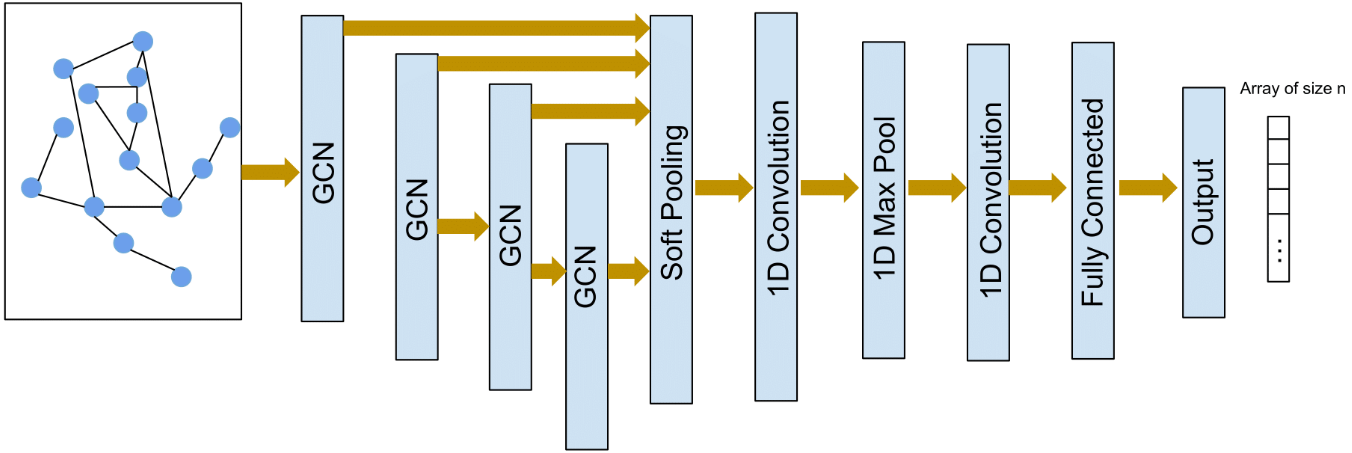
*(138) BLOCK CASE CLASS COND DEFAULT DO_WHILE ELSE FIELD FOR IF IMPORT IMPORTS INIT METHOD MODIFIER NAME PACKAGE PARAMS RETURN ROOT SWITCH Statement THEN TYPE UPDATE VAR WHILE

Relationship Types

*(221) False GO_TO Simple True flows

Property Keys

id label line method normalized type


GCN
GCN
GCN
GCN
Soft Pooling
1D Convolution
1D Max Pool
1D Convolution
Fully Connected
Output
Array of size n
⋮

Benchmark

Combination	F1 score	Loss
AST	0.9286	0.1970
CFG	0.9252	0.1845
PDG	0.9286	0.1981
AST-CFG	0.9269	0.2422
AST-PDG	0.9362	0.2059
CFG-PDG	0.9277	0.2079
APC	0.9736	0.1290

Thanks!

Does anyone have any questions?
Maykel.mattar@univ-ubs.fr