

Generalized API checkers for the Clang Static Analyzer

Sean Eveson
EuroLLVM 27/03/17



Background



- We have custom Static Analyzer checkers for our APIs
- Some are adapted, some are new
- These checkers were initially hard-coded
- We've generalized them to read a YAML description of any API

Unchecked Return Checker



```
ErrorCode initializeAPI();
```

```
int main() {  
    initializeAPI();  
}
```

```
$ clang --analyze example.cpp  
-Xanalyzer -analyzer-config  
-Xanalyzer api-data-path=apidata
```

Warning: The return value from the call to
'initializeAPI' is not checked

Unchecked Return YAML



```
- "initializeAPI"  
- "function1"  
- "function2"  
- "function3"
```

Async Argument Checker



```
void getNumber(int *p, bool async=false);
```

```
int foo() {  
    int x;  
    getNumber(&x, true);  
    ...  
    return x;  
}
```

Warning: Use of stack memory 'x' in call to
'getNumber' is dangerous

Async Argument YAML



```
simple:  
  NonStack: [1]  
  
multiple:  
  NonStack: [1, 3, 4]  
  
getNumber:  
  NonStack: [0]  
  BoolConditions:  
    - {Param: 1, Value: true}
```

Argument Value Checker



```
void minusTenToTen(int x);

int main() {
    bar(-8, 4);
}

int bar(int a, int b) {
    minusTenToTen(a - b);
}
```

Warning: The 1st argument of function
'minusTenToTen' must be in the range
[-10,10]

Argument Value YAML



```
minusTenToTen:
  - Param: 0
    Type: IN
    Value: [-10, 10]

greaterThanFive:
  - Param: 2
    Type: GT
    Value: [5]

notEqualMinusOne:
  - Param: 1
    Type: NEQ
    Value: [-1]
```

Further Work



We aim to upstream these checkers so anyone can check any
APIs with minimal effort.