PVS-Studio

Integrating SAST into DevSecOps

PVS-Studio

Anton Tretyakov
tretyakov@viva64.com

In today’s program

In today's program

- What it is

In today’s program

- What it is
- How it works

In today’s program

- What it is
- How it works
- How to DevSecOps it

In today’s program

- What it is
- How it works
- How to DevSecOps it
- Q/A

Anton Tretyakov

C++ engineer

- Develop PVS-Studio tools
- Maintain infrastructure
- Write articles about C++

What is SAST

What is SAST

Baby don’t hurt me

What is SAST?

What is SAST?

- SAST is a code check

What is SAST?

- SAST is an automated code check

What is SAST?

- SAST is an automated code check
- without its execution

What is SAST?

- SAST is an automated code check
- without its execution
- for potential vulnerabilities

What is SAST?

- SAST is an automated code check
- without its execution
- for potential vulnerabilities

What is vulnerability?

Bug

```
void bad_func(int a)
{
   if(a == 0) {
      42 / 0;
   }
}
```

Bug

Potential Vuln

CWE-369: Divide By Zero

Technical Impact: *DoS: Crash, Exit, or Restart*

A Divide by Zero results in a crash.

Likelihood Of Exploit: Medium

Bug

Potential
Vuln

Real
Vuln

CVE-2007-3268

Invalid size value leads to divide by zero.

CVE-2007-2723

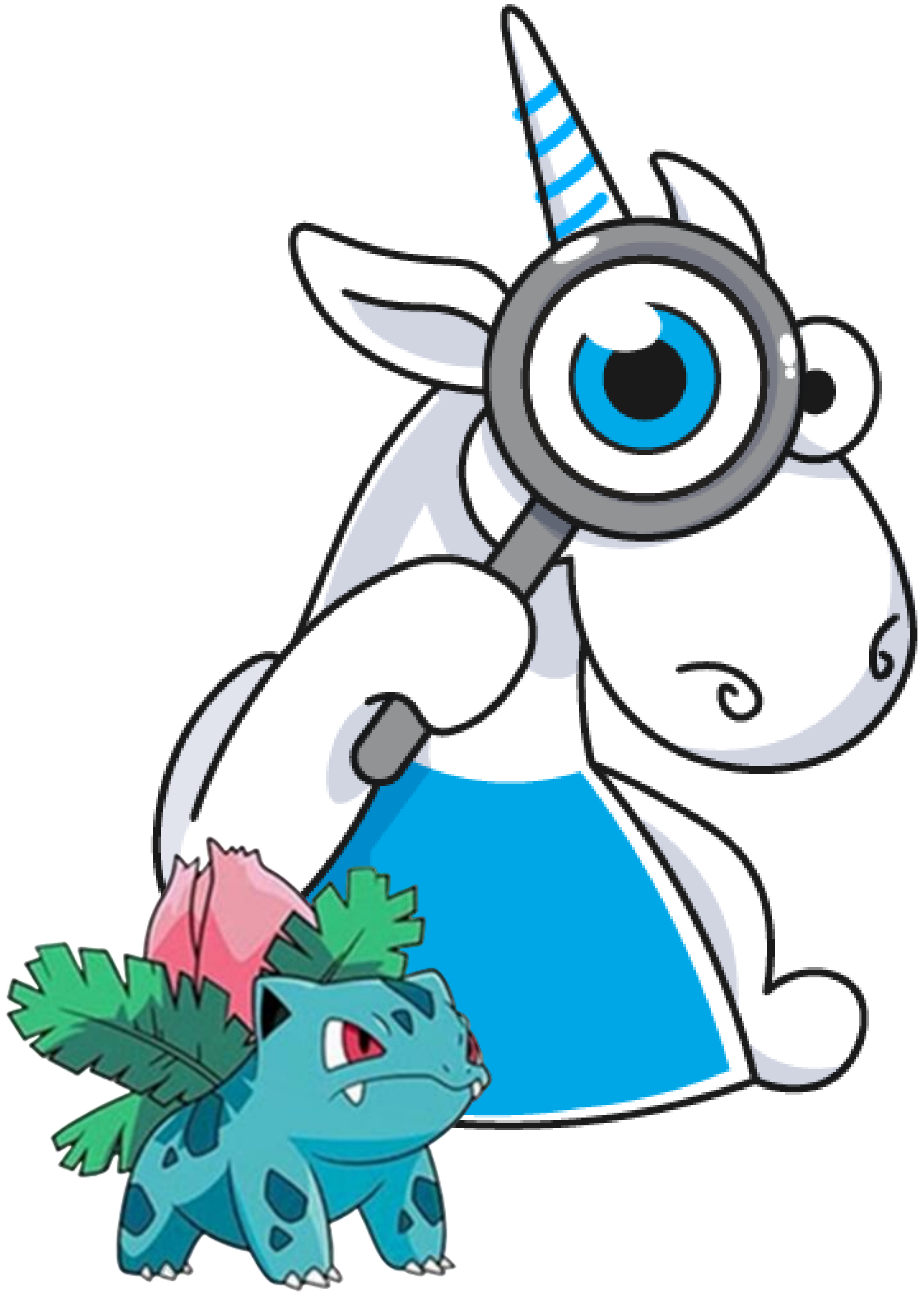
"Empty" content triggers divide by zero.

CVE-2007-2237

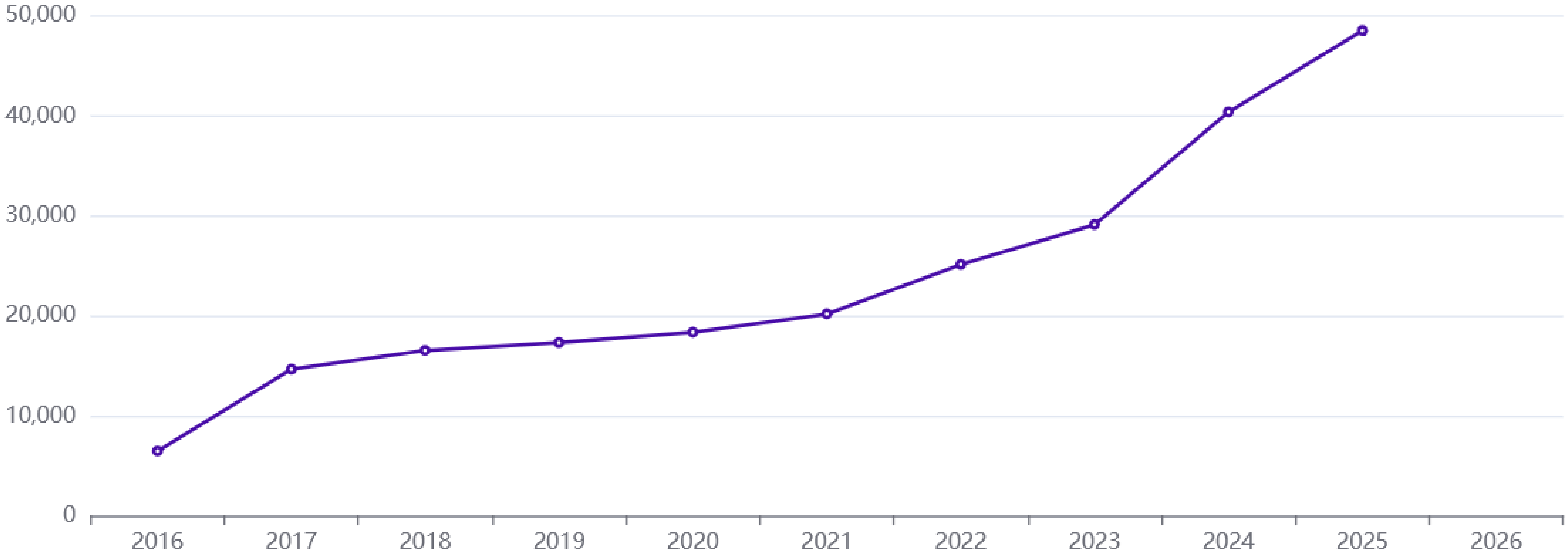
Height value of 0 triggers divide by zero.

CVE-2007-2237

Microsoft Windows Graphics Device Interface (GDI+, GdiPlus.dll) allows context-dependent attackers to cause a denial of service (crash) via an ICO file with an InfoHeader containing a Height of zero, which triggers a divide-by-zero error.

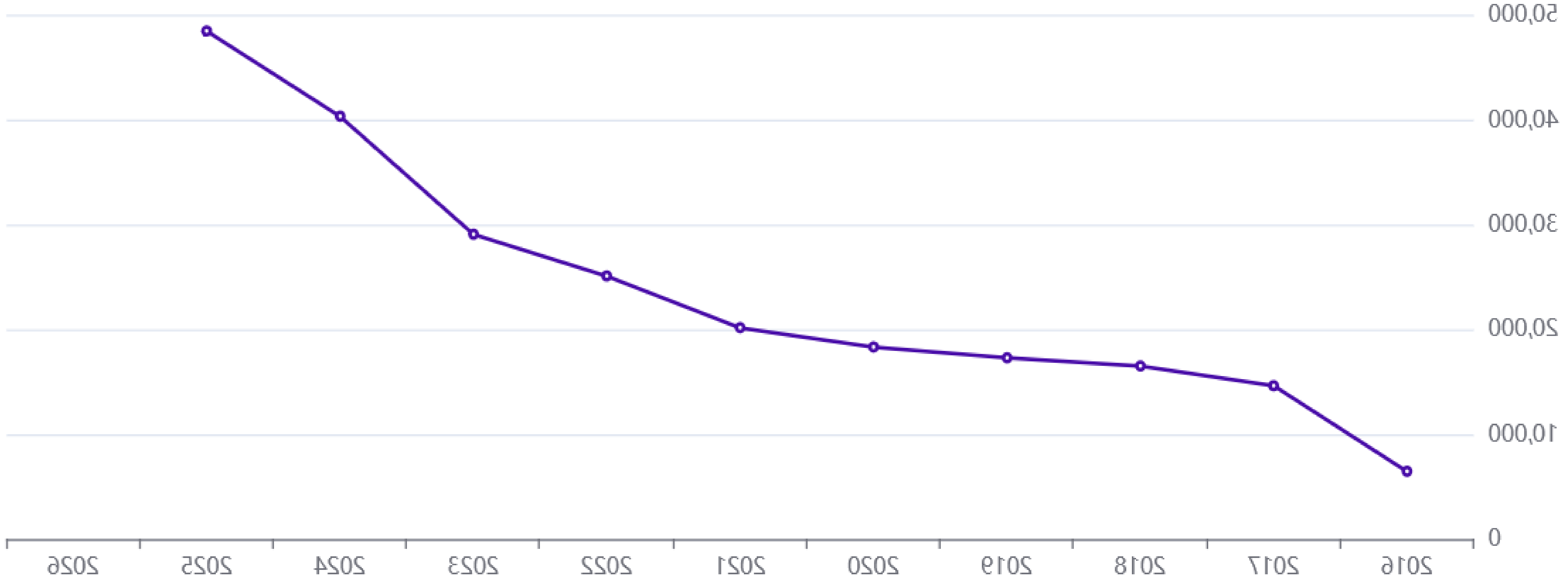


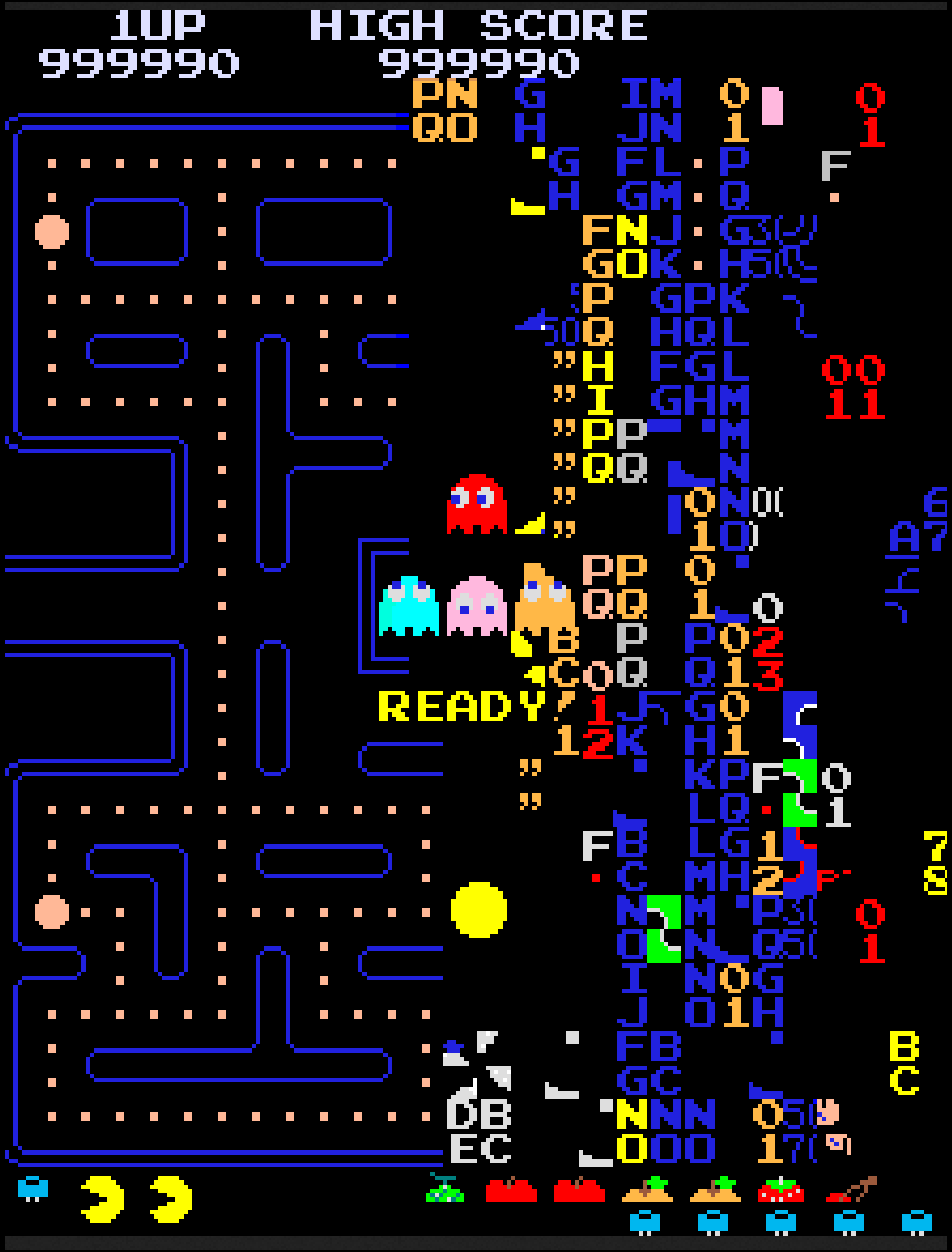
Vulns by year

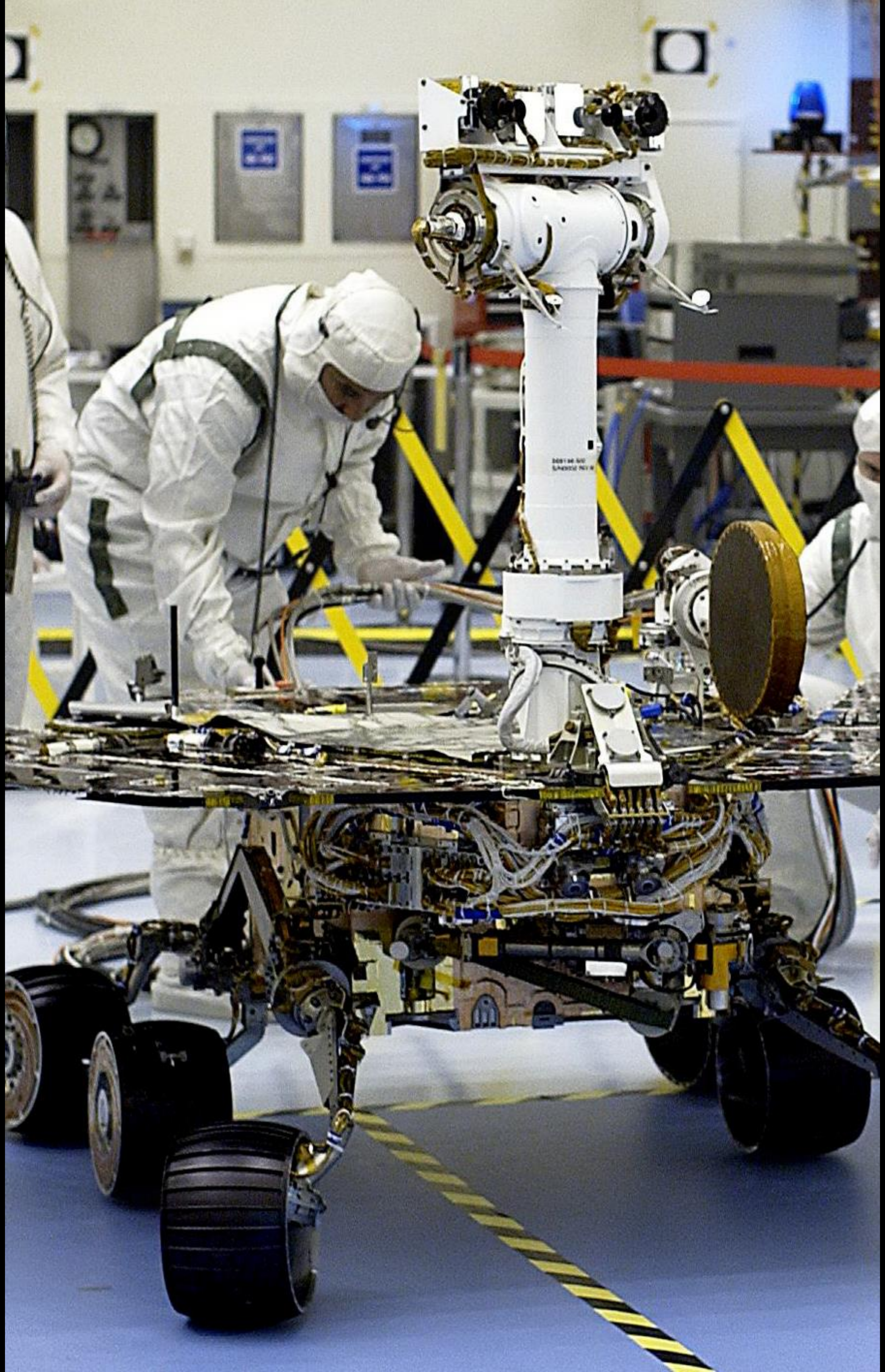


https://www.cvedetails.com

Desired outcome



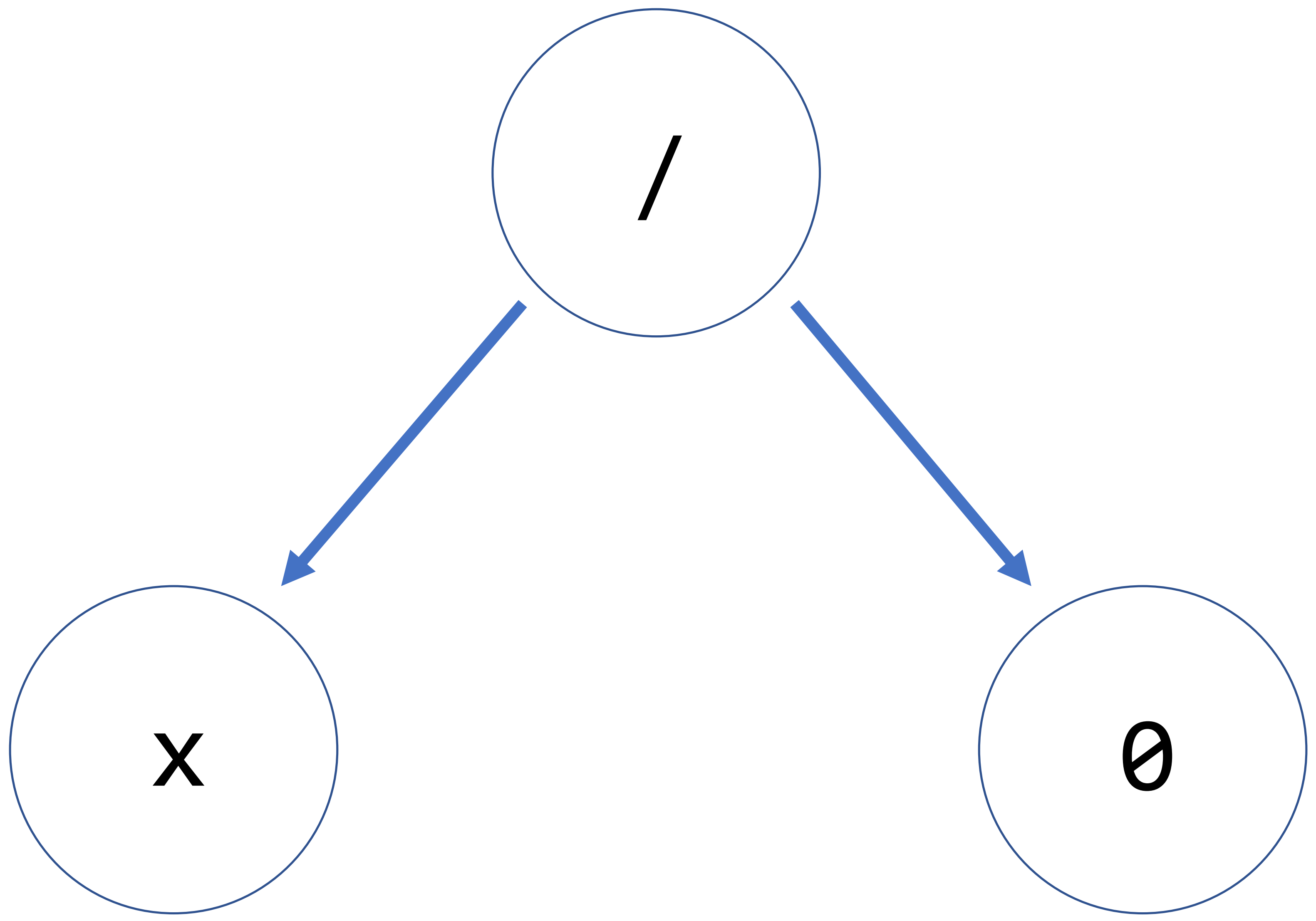

1UP
999990
HIGH SCORE
999990
READY!

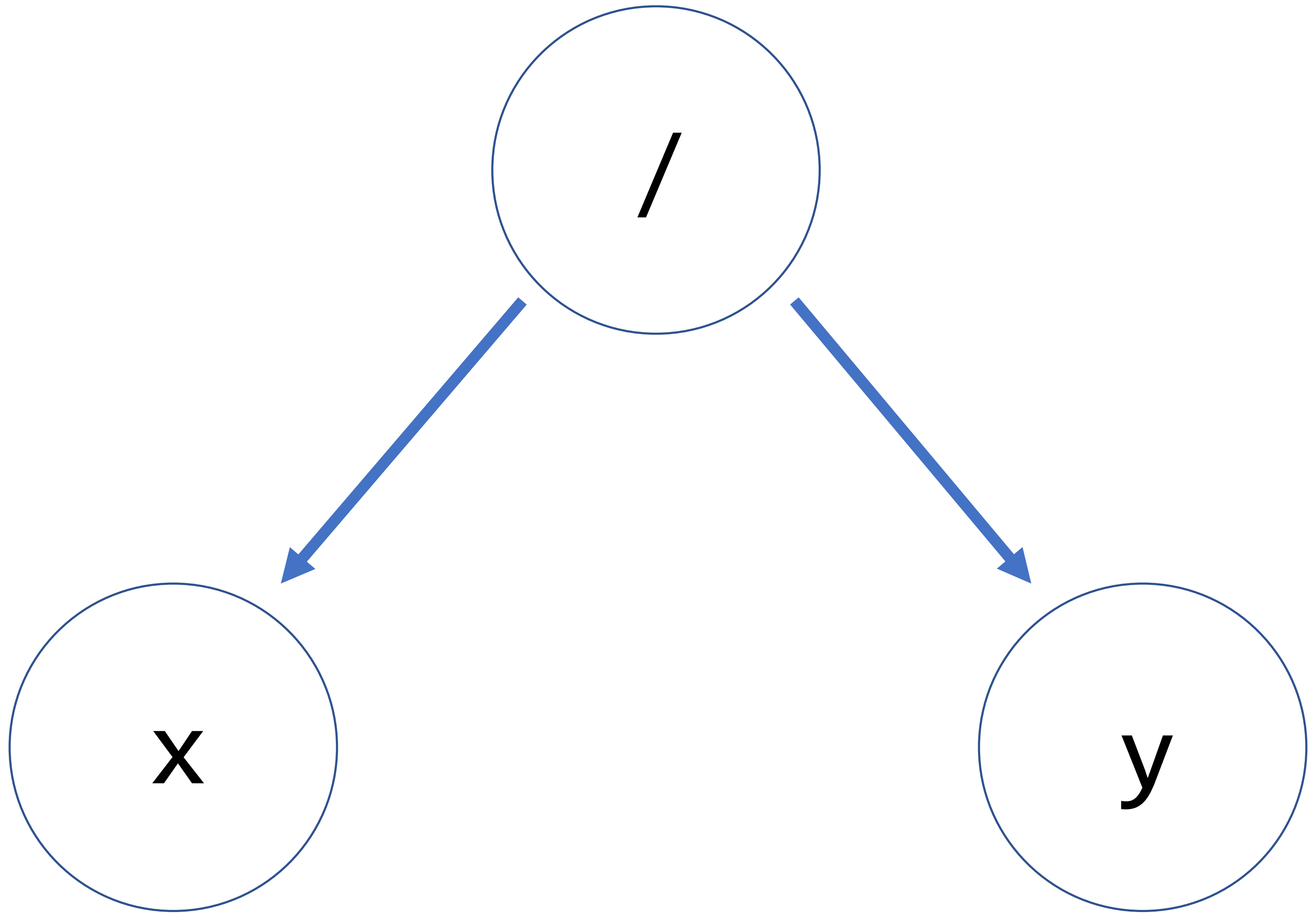


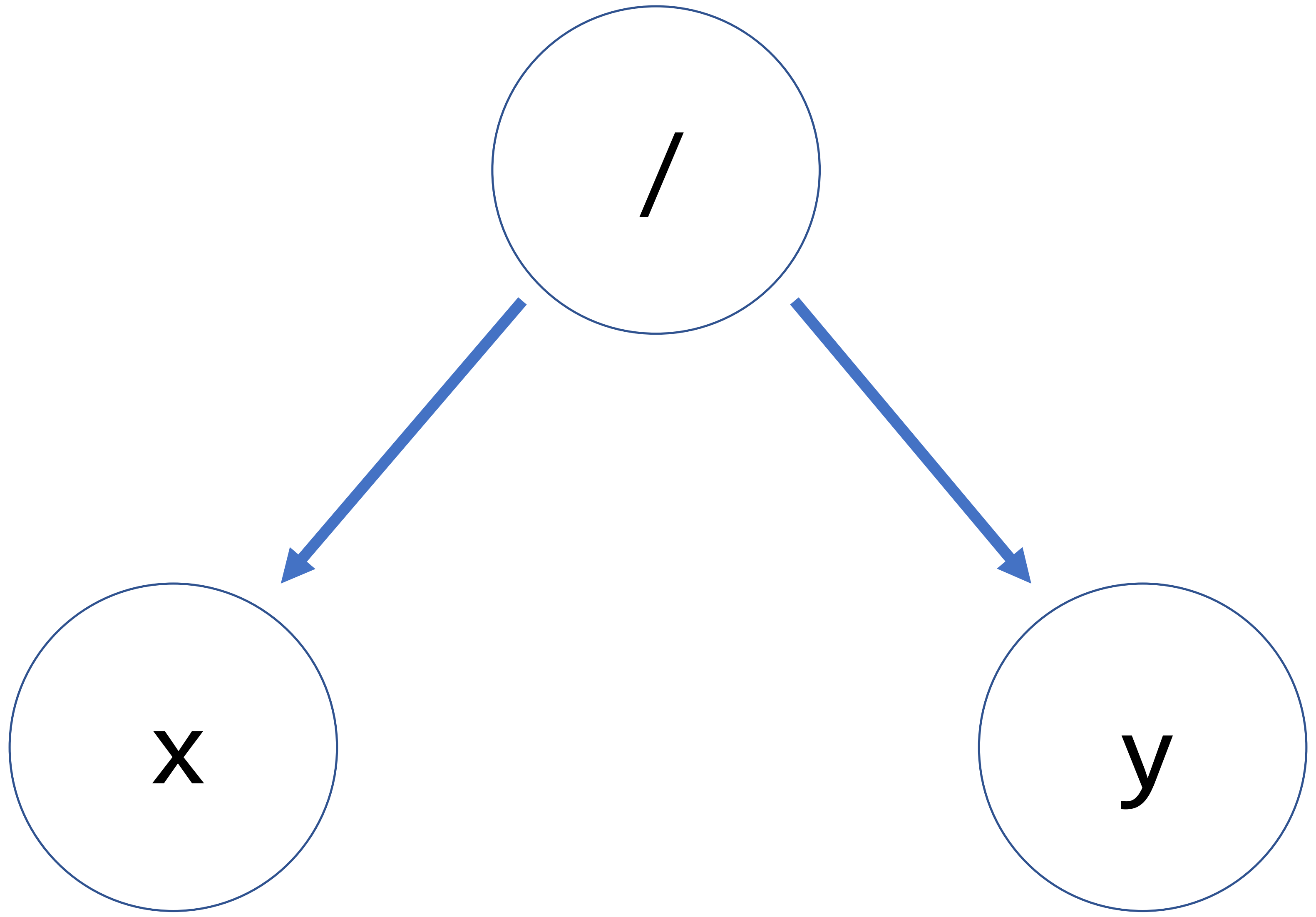
How SAST works

How to find div by zero ?

```
if some.Expr.contains(“/ 0”) then raise Error
if some.Expr.contains(“/ 0”) then raise Error

if some.Expr.contains(“/0 ”) then raise Error
if some.Expr.contains(“/ 0”) then raise Error

if some.Expr.contains(“/0 ”) then raise Error

if some.Expr.contains(“/    0”) then raise Error
```

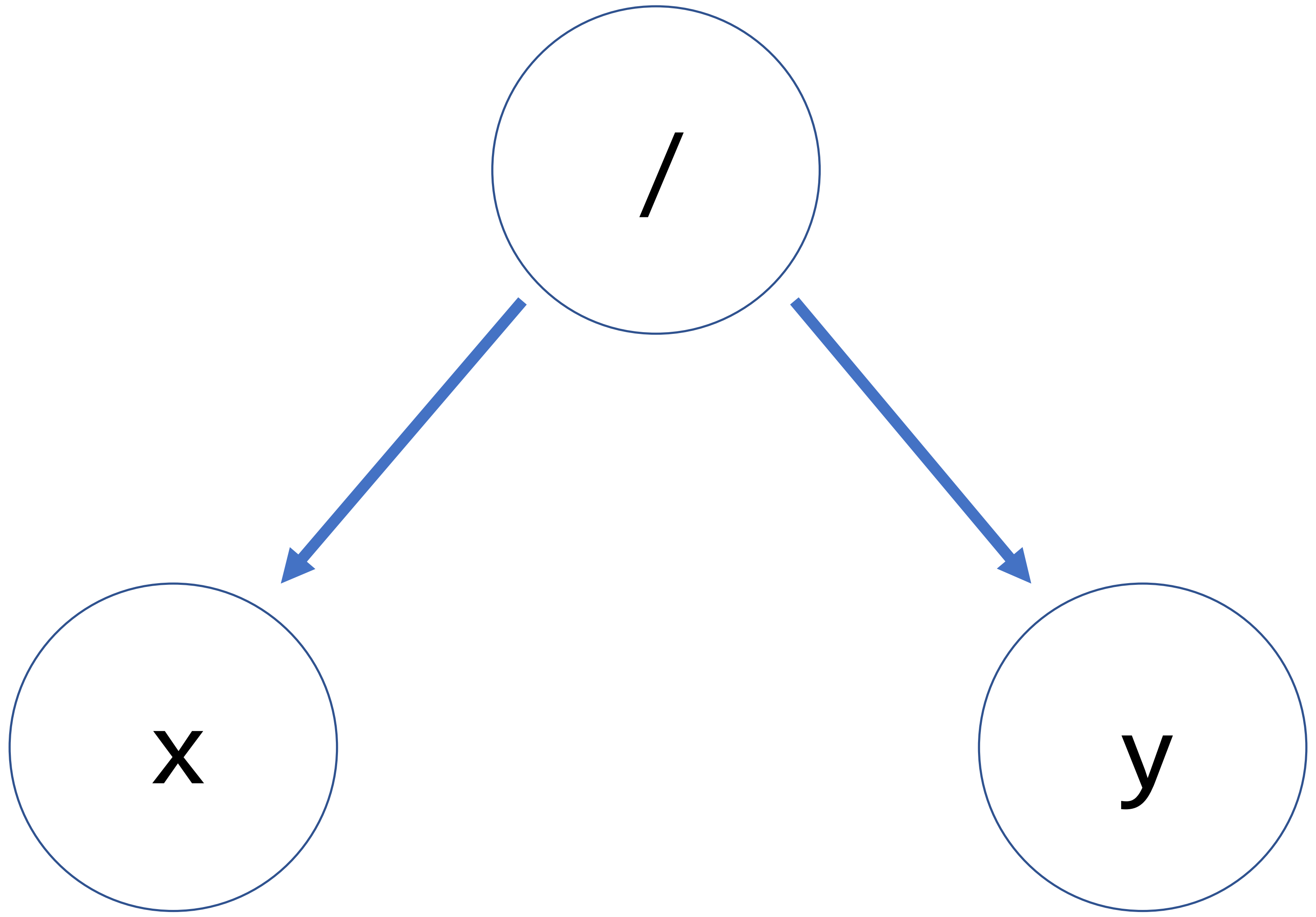

/
x
0


/
x
y



Type: s_int_64
Range: [100, 256)

Type: s_int_64
Range: [-30, 12)



Type: s_int_64
Range: [100, 256)

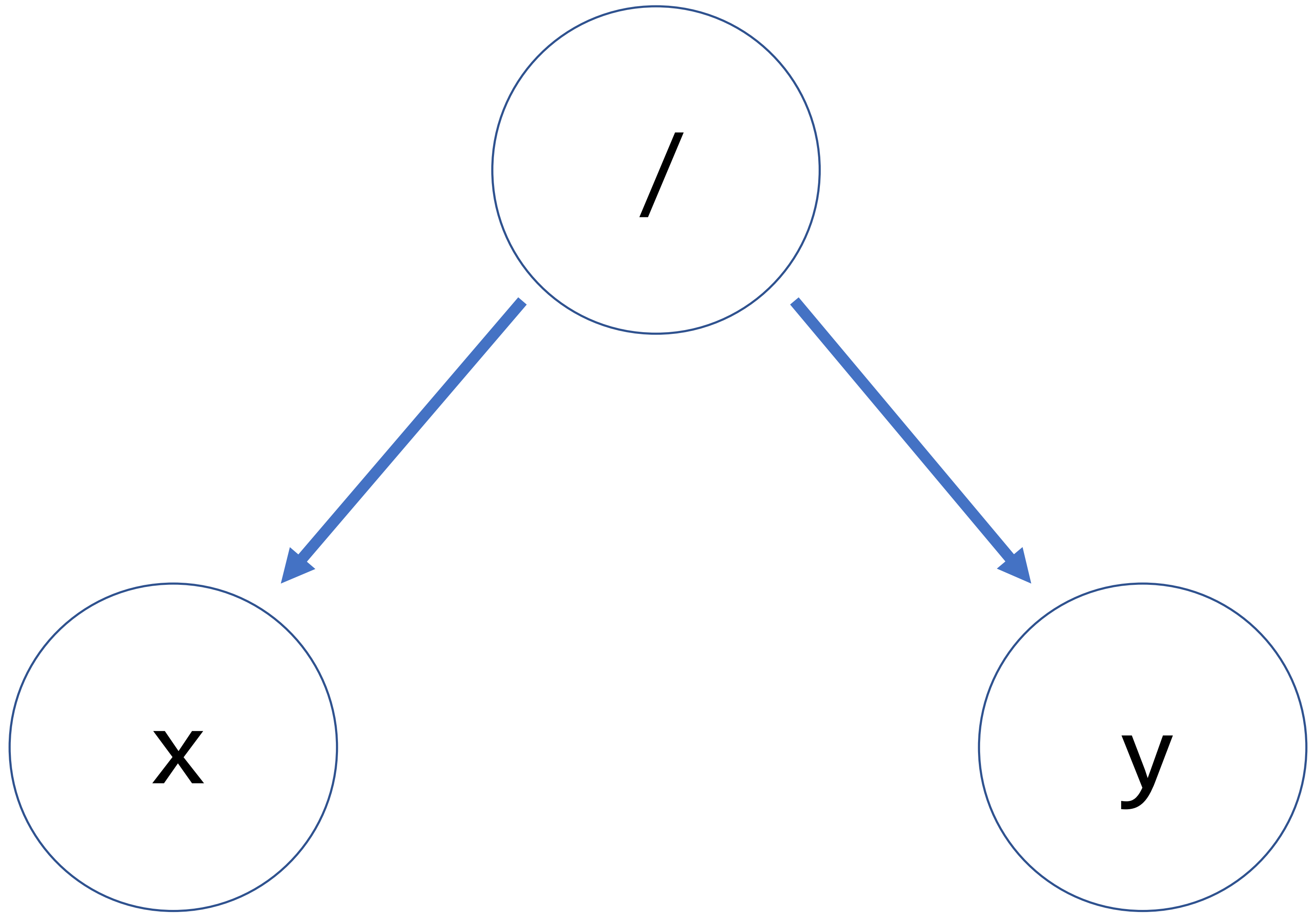
Type: s_int_64
Range: [-30, 12)

Possible division by zero!

```
const char *
TGHtml::GetPctWidth(TGHtmlElement *p, char *opt, char *ret)
{
  int n, m, val;
  //...
  if (n < 0 || n > 100) return z;
  //...
  if (!fInTd) {
    snprintf(ret, 15, "%d", val / n);
  }
  //...
}
```

```
const char *
TGHtml::GetPctWidth(TGHtmlElement *p, char *opt, char *ret)
{
  int n, m, val;
  //...
  if (n < 0 || n > 100) return z;
  //...
  if (!fInTd) {
    snprintf(ret, 15, "%d", val / n);
  }
  //...
}
```

PVS-Studio warning: Divide by zero. Denominator range [0..100].



Type: s_int_64
Range: [100, 256)

Type: s_int_64
Range: [-30, 12)

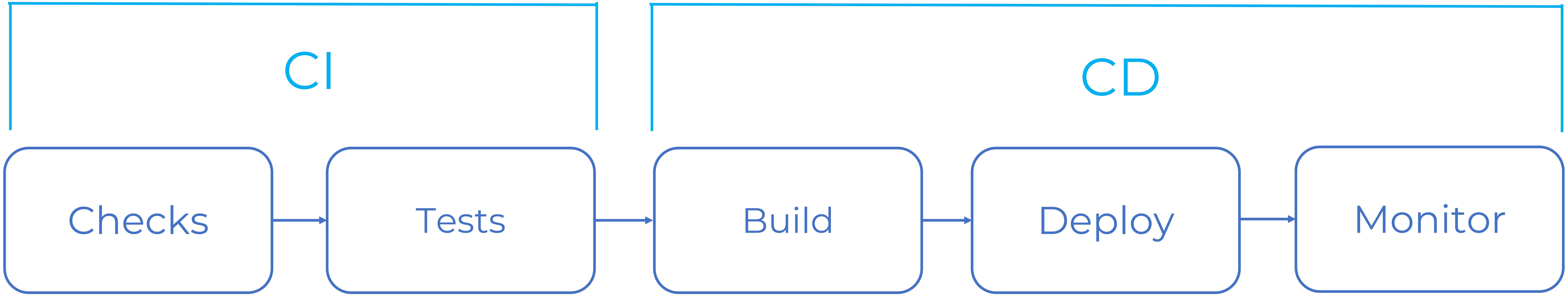
```
struct HeifFrameInfo
{
  //...
  void set(/*...*/) {
    //...
    mIccData.reset(new uint8_t[iccSize]);
    //...
  }
  //...
  std::unique_ptr<uint8_t> mIccData;
};
```

```
struct HeifFrameInfo
{
  //...
  void set(/*...*/) {
    //...
    mIccData.reset(new uint8_t[iccSize]);
    //...
  }
  //...
  std::unique_ptr<uint8_t> mIccData;
};
```

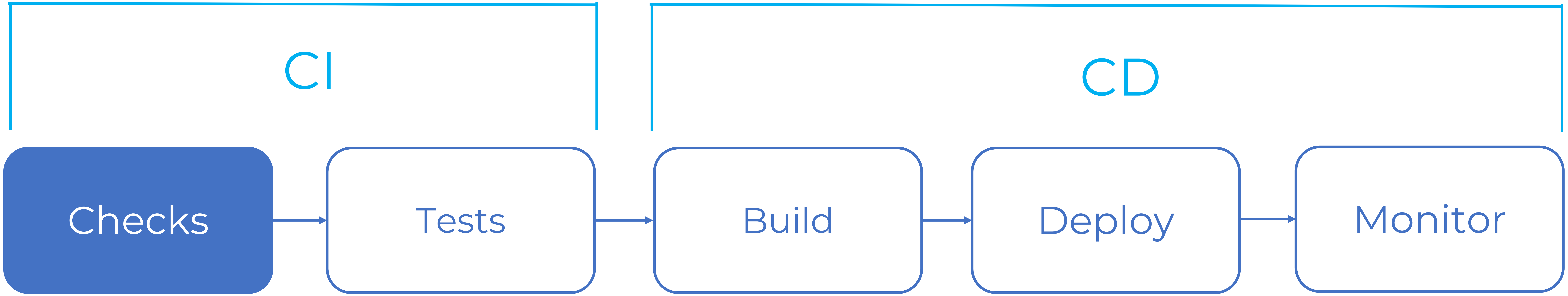
PVS-Studio warning: Incorrect use of unique_ptr.

How to DevSecOps SAST

Where SAST fits?



Where SAST fits?



SAST as quality gate

```yaml
image: registry.gitlab.com/pvs-studio/pvs_studio_demo_cs_project

workflow:
  rules:
    - if: $CI_PIPELINE_SOURCE == 'merge_request_event'

stages:
  - build
  - analyze

build:
  stage: build
  script:
    - 'dotnet build'

analyze:
  stage: analyze
  needs: [build]
  dependencies: []
  script:
    - 'pvs-studio-analyzer credentials $PVS_STUDIO_LICENSE_NAME $PVS_STUDIO_LICENSE_KEY'
    - 'pvs-studio-dotnet -t dotnetcore.csproj -o pvs.json || :'
    - 'plog-converter -t totals -a all -o pvs_metrics.txt pvs.json'
  artifacts:
    expose_as: 'PVS-Studio report'
    paths: ['pvs_metrics.txt']

quality_gate:
  stage: analyze
  needs: [analyze]
  dependencies: [analyze]
  script:
    - read PVS_HIGH_LEVEL_COUNT PVS_MID_LEVEL_COUNT < <(grep "Total L1:" pvs_metrics.txt | sed -E 's/.*L1:([0-9]+) \+ L2:([0-9]+).*/\1 \2/')
    - '[ "$PVS_HIGH_LEVEL_COUNT" -gt "$PVS_MAX_HIGH_LEVEL_COUNT" ] && { exit 1; } || :'
    - '[ "$PVS_MID_LEVEL_COUNT"  -gt "$PVS_MAX_MID_LEVEL_COUNT"  ] && { exit 1; } || :'
```

```
image: registry.gitlab.com/pvs-studio/pvs_studio_demo_cs_project

workflow:
  rules:
    - if: $CI_PIPELINE_SOURCE == 'merge_request_event'

stages:
  - build
  - analyze

build:
  stage: build
  script:
    - 'dotnet build’
```

```
analyze:
  stage: analyze
  needs: [build]
  dependencies: []
  script:
    - 'pvs-studio-analyzer credentials $LICENSE_NAME $LICENSE_KEY'
    - 'pvs-studio-dotnet -t dotnetcore.csproj -o pvs.json || :'
    - 'plog-converter -t totals -a all -o pvs_metrics.txt pvs.json'
  artifacts:
    expose_as: 'PVS-Studio report'
    paths: ['pvs_metrics.txt']
```

```
analyze:
  stage: analyze
  needs: [build]
  dependencies: []
  script:
    - 'pvs-studio-analyzer credentials $LICENSE_NAME $LICENSE_KEY'
    - 'pvs-studio-dotnet -t dotnetcore.csproj -o pvs.json || :'
    - 'plog-converter -t totals -a all -o pvs_metrics.txt pvs.json'
  artifacts:
    expose_as: 'PVS-Studio report'
    paths: ['pvs_metrics.txt']
```

```
analyze:
  stage: analyze
  needs: [build]
  dependencies: []
  script:
    - 'pvs-studio-analyzer credentials $LICENSE_NAME $LICENSE_KEY'
    - 'pvs-studio-dotnet -t dotnetcore.csproj -o pvs.json || :'
    - 'plog-converter -t totals -a all -o pvs_metrics.txt pvs.json'
  artifacts:
    expose_as: 'PVS-Studio report'
    paths: ['pvs_metrics.txt']
```

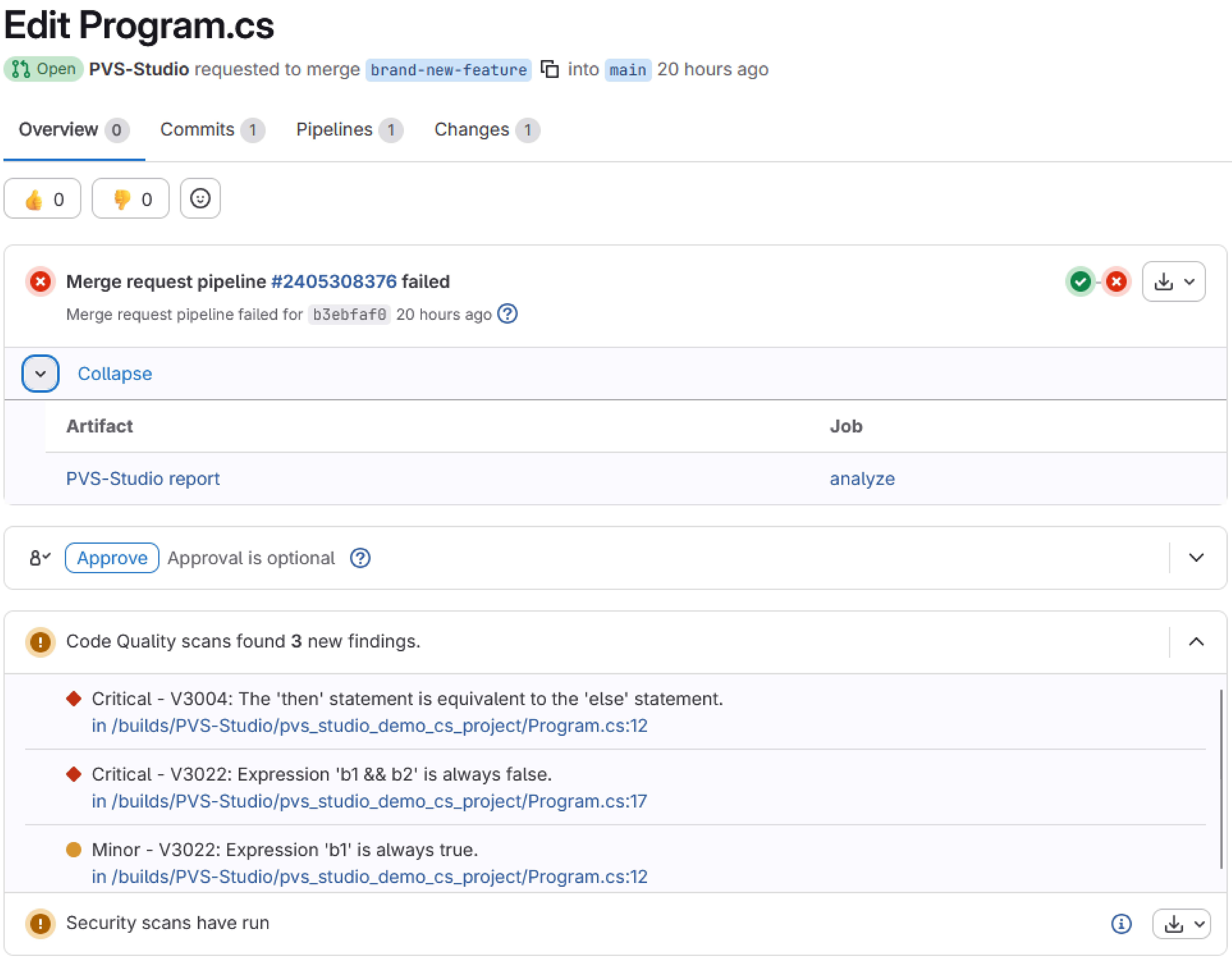
```
analyze:
  stage: analyze
  needs: [build]
  dependencies: []
  script:
    - 'pvs-studio-analyzer credentials $LICENSE_NAME $LICENSE_KEY'
    - 'pvs-studio-dotnet -t dotnetcore.csproj -o pvs.json || :'
    - 'plog-converter -t totals -a all -o pvs_metrics.txt pvs.json'
  artifacts:
    expose_as: 'PVS-Studio report'
    paths: ['pvs_metrics.txt']
```

```
analyze:
  stage: analyze
  needs: [build]
  dependencies: []
  script:
    - 'pvs-studio-analyzer credentials $LICENSE_NAME $LICENSE_KEY'
    - 'pvs-studio-dotnet -t dotnetcore.csproj -o pvs.json || :'
    - 'plog-converter -t totals -a all -o pvs_metrics.txt pvs.json'
  artifacts:
    expose_as: 'PVS-Studio report'
    paths: ['pvs_metrics.txt']
```

```
quality_gate:
  stage: analyze
  needs: [analyze]
  dependencies: [analyze]
  script:
    - read PVS_HIGH_LEVEL_COUNT PVS_MID_LEVEL_COUNT < <(grep "Total L1:"
pvs_metrics.txt | sed -E 's/.*L1:([0-9]+) \+ L2:([0-9]+).*/\1 \2/')
    - '[ "$PVS_HIGH_LEVEL_COUNT" -gt "$PVS_MAX_HIGH_LEVEL_COUNT" ] && { exit 1; } ||
:'
    - '[ "$PVS_MID_LEVEL_COUNT"  -gt "$PVS_MAX_MID_LEVEL_COUNT"  ] && { exit 1; } ||
:’
```

```
quality_gate:
  stage: analyze
  needs: [analyze]
  dependencies: [analyze]
  script:
    - read PVS_HIGH_LEVEL_COUNT PVS_MID_LEVEL_COUNT < <(grep "Total L1:"
pvs_metrics.txt | sed -E 's/.*L1:([0-9]+) \+ L2:([0-9]+).*/\1 \2/')
    - '[ "$PVS_HIGH_LEVEL_COUNT" -gt "$PVS_MAX_HIGH_LEVEL_COUNT" ] && { exit 1; } ||
:'
    - '[ "$PVS_MID_LEVEL_COUNT"  -gt "$PVS_MAX_MID_LEVEL_COUNT"  ] && { exit 1; } ||
:’
```

```
quality_gate:
  stage: analyze
  needs: [analyze]
  dependencies: [analyze]
  script:
    - read PVS_HIGH_LEVEL_COUNT PVS_MID_LEVEL_COUNT < <(grep "Total L1:"
pvs_metrics.txt | sed -E 's/.*L1:([0-9]+) \+ L2:([0-9]+).*/\1 \2/')
    - '[ "$PVS_HIGH_LEVEL_COUNT" -gt "$PVS_MAX_HIGH_LEVEL_COUNT" ] && { exit 1; } ||
:'
    - '[ "$PVS_MID_LEVEL_COUNT"  -gt "$PVS_MAX_MID_LEVEL_COUNT"  ] && { exit 1; } ||
:’
```



Edit Program.cs

Open **PVS-Studio** requested to merge `brand-new-feature` into `main` 20 hours ago

Merge request pipeline #2405308376 failed

Merge request pipeline failed for `b3ebfaf0` 20 hours ago

Artifact	Job
PVS-Studio report	analyze

Approval is optional

Code Quality scans found **3** new findings.

- Critical - V3004: The 'then' statement is equivalent to the 'else' statement.
  in /builds/PVS-Studio/pvs_studio_demo_cs_project/Program.cs:12
- Critical - V3022: Expression 'b1 && b2' is always false.
  in /builds/PVS-Studio/pvs_studio_demo_cs_project/Program.cs:17
- Minor - V3022: Expression 'b1' is always true.
  in /builds/PVS-Studio/pvs_studio_demo_cs_project/Program.cs:12

Security scans have run

Incremental checks

```
get_file_changes:
    stage: analyze
    script:
      - |
      if [ "$CI_PIPELINE_SOURCE" = "merge_request_event" ]; then
        BASE=$CI_MERGE_REQUEST_TARGET_BRANCH_SHA
      else
        BASE=$CI_COMMIT_BEFORE_SHA
        if [ "$BASE" = "0000000000000000000000000000000000000000" ]; then
          BASE=$(git rev-parse HEAD~1)
        fi
      fi
    - git diff-tree --no-commit-id --name-only \
      -r $BASE -r $CI_COMMIT_SHA -- "*.cs" > changed_files.txt
    artifacts:
        paths:
          - changed_files.txt
```

```
get_file_changes:
    stage: analyze
    script:
      - |
      if [ "$CI_PIPELINE_SOURCE" = "merge_request_event" ]; then
        BASE=$CI_MERGE_REQUEST_TARGET_BRANCH_SHA
      else
        BASE=$CI_COMMIT_BEFORE_SHA
        if [ "$BASE" = "0000000000000000000000000000000000000000" ]; then
          BASE=$(git rev-parse HEAD~1)
        fi
      fi
    - git diff-tree --no-commit-id --name-only \
      -r $BASE -r $CI_COMMIT_SHA -- "*.cs" > changed_files.txt
    artifacts:
        paths:
          - changed_files.txt
```

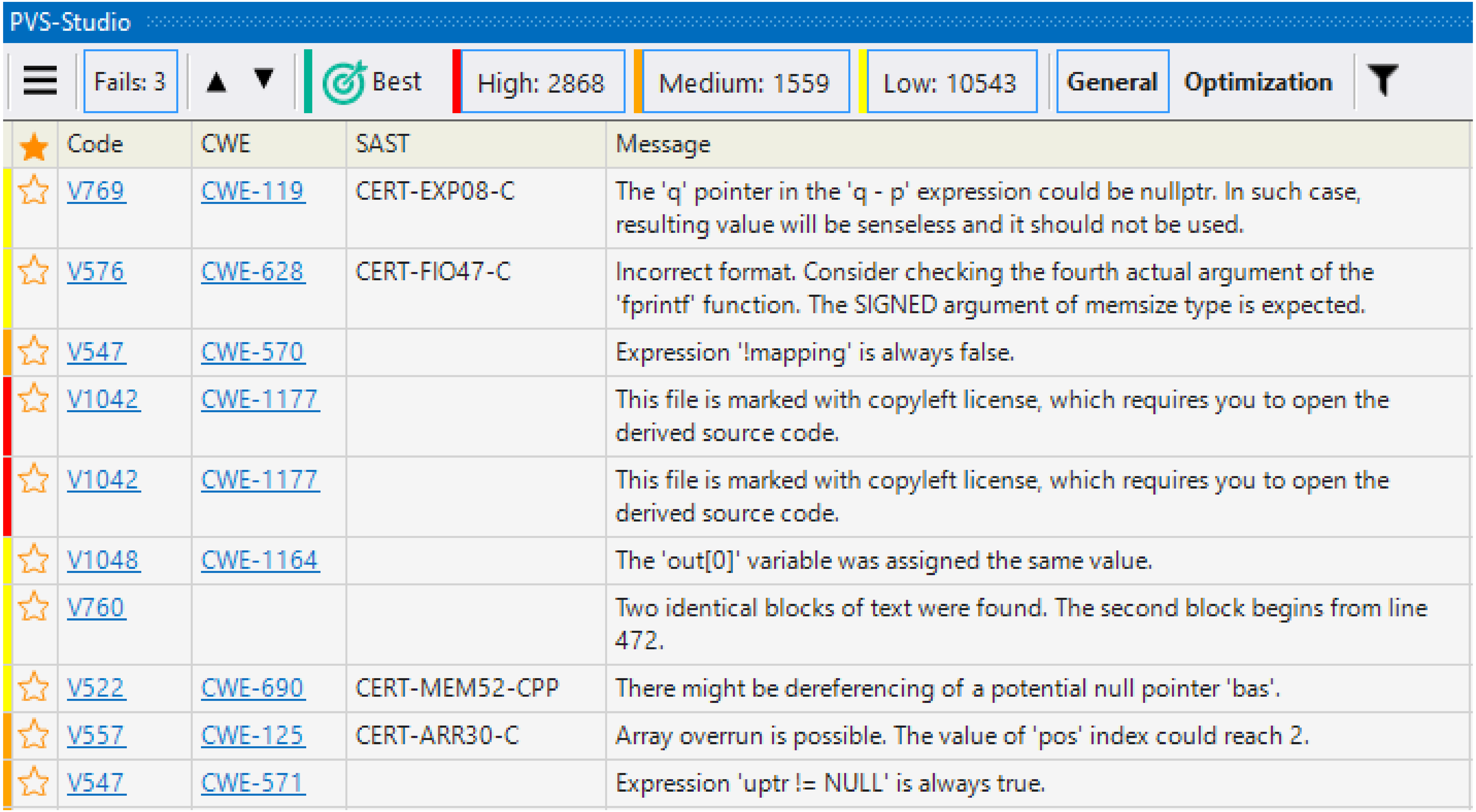
```
get_file_changes:
    stage: analyze
    script:
      - |
      if [ "$CI_PIPELINE_SOURCE" = "merge_request_event" ]; then
        BASE=$CI_MERGE_REQUEST_TARGET_BRANCH_SHA
      else
        BASE=$CI_COMMIT_BEFORE_SHA
        if [ "$BASE" = "0000000000000000000000000000000000000000" ]; then
          BASE=$(git rev-parse HEAD~1)
        fi
      fi
    - git diff-tree --no-commit-id --name-only \
      -r $BASE -r $CI_COMMIT_SHA -- "*.cs" > changed_files.txt
    artifacts:
        paths:
          - changed_files.txt
```

```
analyze:
  stage: analyze
  needs: [build, get_file_changes]
  dependencies: [get_file_changes]
  script:
    - 'pvs-studio-analyzer credentials $LICENSE_NAME $LICENSE_KEY'
    - 'pvs-studio-dotnet -t dotnetcore.csproj -f changed_files.txt -o pvs.json || :'
    - 'plog-converter -t totals -a all -o pvs_metrics.txt pvs.json'
  artifacts:
    expose_as: 'PVS-Studio report'
    paths: ['pvs_metrics.txt']
```

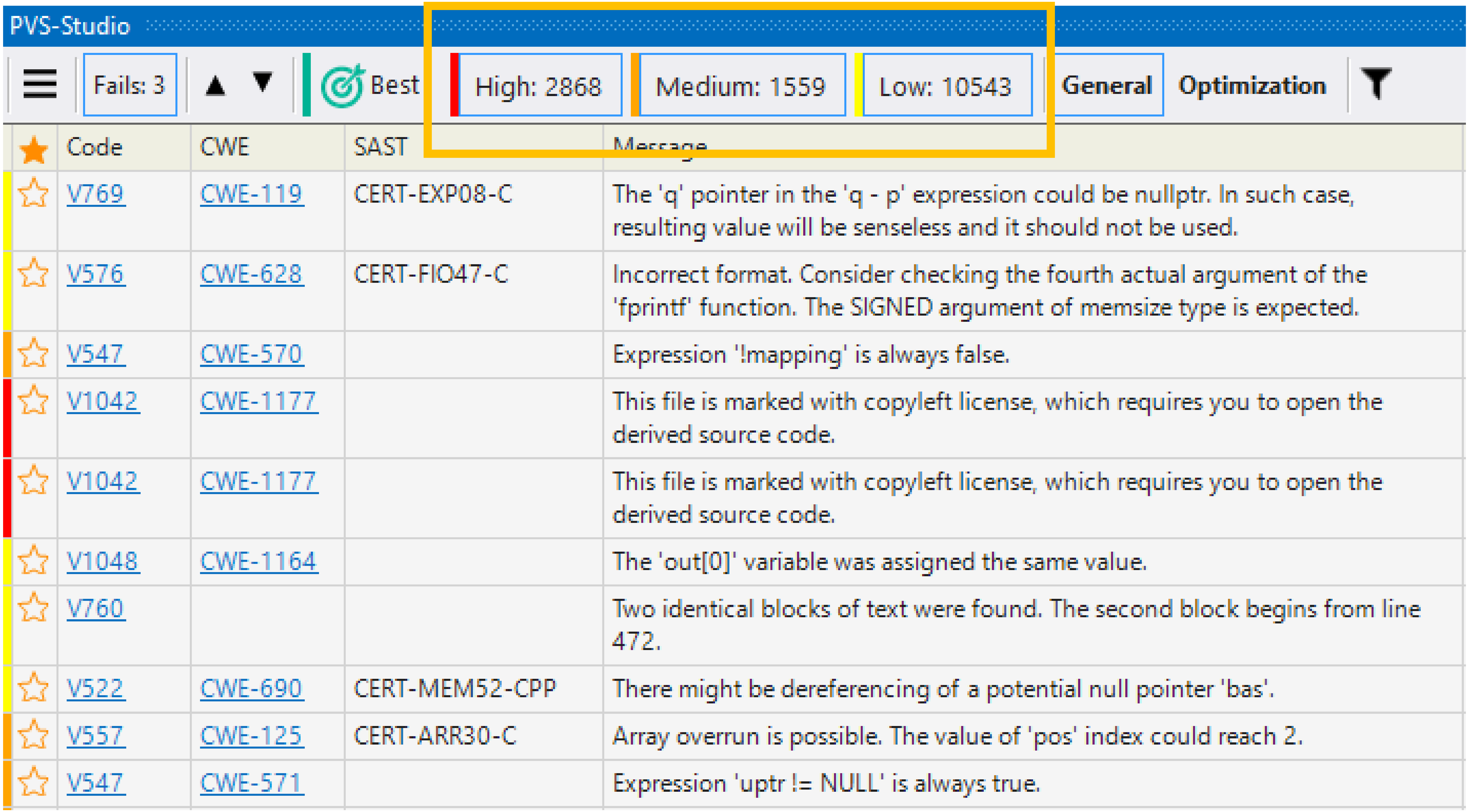
```
analyze:
  stage: analyze
  needs: [build, get_file_changes]
  dependencies: [get_file_changes]
  script:
    - 'pvs-studio-analyzer credentials $LICENSE_NAME $LICENSE_KEY'
    - 'pvs-studio-dotnet -t dotnetcore.csproj -f changed_files.txt -o pvs.json || :'
    - 'plog-converter -t totals -a all -o pvs_metrics.txt pvs.json'
  artifacts:
    expose_as: 'PVS-Studio report'
    paths: ['pvs_metrics.txt']
```

```
analyze:
  stage: analyze
  needs: [build, get_file_changes]
  dependencies: [get_file_changes]
  script:
    - 'pvs-studio-analyzer credentials $LICENSE_NAME $LICENSE_KEY'
    - 'pvs-studio-dotnet -t dotnetcore.csproj -f changed_files.txt -o pvs.json || :'
    - 'plog-converter -t totals -a all -o pvs_metrics.txt pvs.json'
  artifacts:
    expose_as: 'PVS-Studio report'
    paths: ['pvs_metrics.txt']
```

What about legacy?

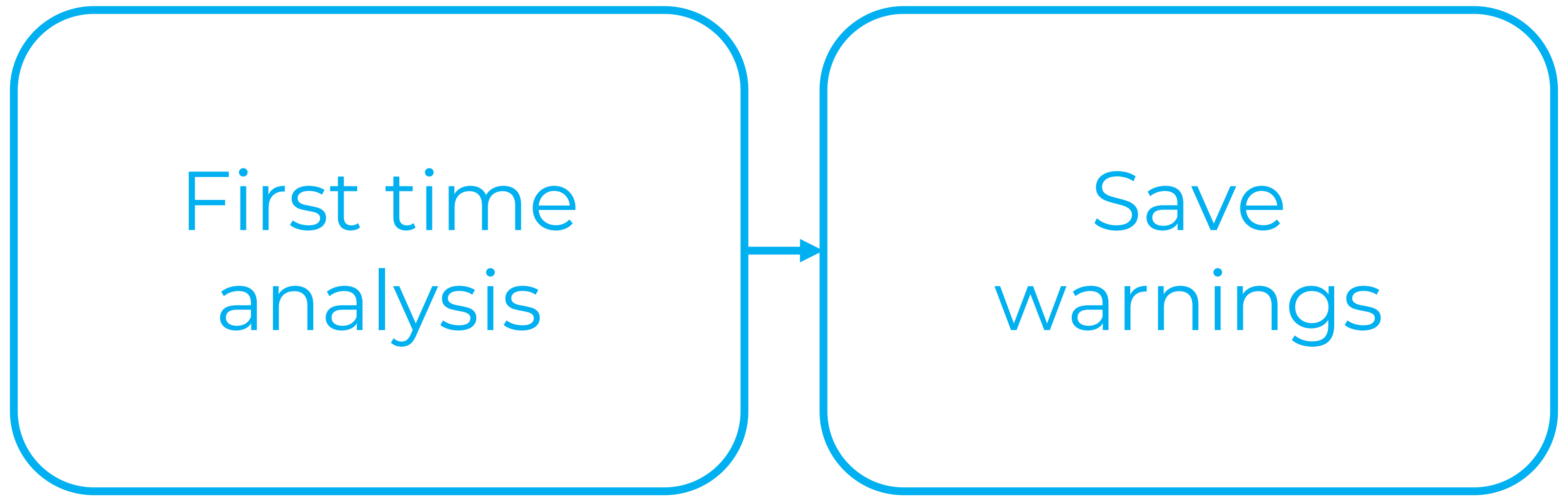


★	Code	CWE	SAST	Message
☆	V769	CWE-119	CERT-EXP08-C	The 'q' pointer in the 'q - p' expression could be nullptr. In such case, resulting value will be senseless and it should not be used.
☆	V576	CWE-628	CERT-FIO47-C	Incorrect format. Consider checking the fourth actual argument of the 'fprintf' function. The SIGNED argument of memsize type is expected.
☆	V547	CWE-570		Expression '!mapping' is always false.
☆	V1042	CWE-1177		This file is marked with copyleft license, which requires you to open the derived source code.
☆	V1042	CWE-1177		This file is marked with copyleft license, which requires you to open the derived source code.
☆	V1048	CWE-1164		The 'out[0]' variable was assigned the same value.
☆	V760			Two identical blocks of text were found. The second block begins from line 472.
☆	V522	CWE-690	CERT-MEM52-CPP	There might be dereferencing of a potential null pointer 'bas'.
☆	V557	CWE-125	CERT-ARR30-C	Array overrun is possible. The value of 'pos' index could reach 2.
☆	V547	CWE-571		Expression 'uptr != NULL' is always true.



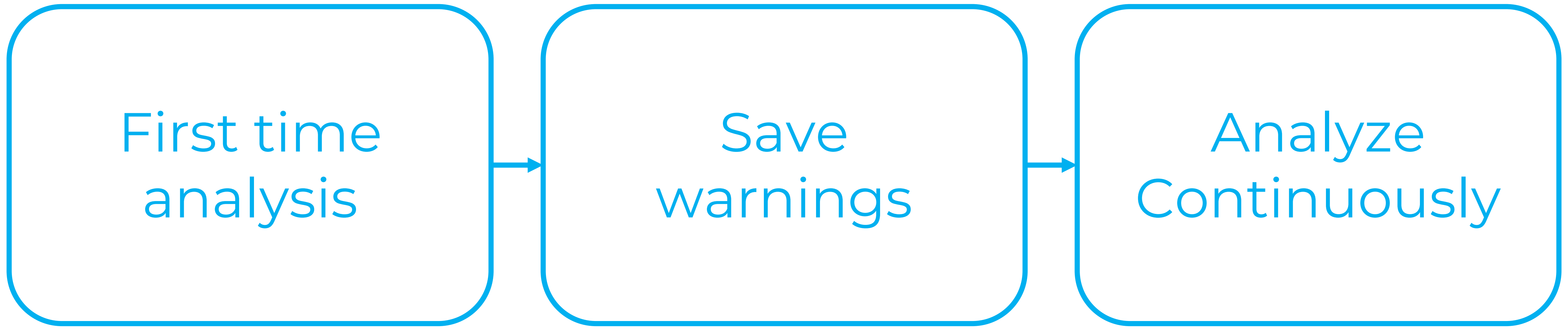
★	Code	CWE	SAST	Message
☆	V769	CWE-119	CERT-EXP08-C	The 'q' pointer in the 'q - p' expression could be nullptr. In such case, resulting value will be senseless and it should not be used.
☆	V576	CWE-628	CERT-FIO47-C	Incorrect format. Consider checking the fourth actual argument of the 'fprintf' function. The SIGNED argument of memsize type is expected.
☆	V547	CWE-570		Expression '!mapping' is always false.
☆	V1042	CWE-1177		This file is marked with copyleft license, which requires you to open the derived source code.
☆	V1042	CWE-1177		This file is marked with copyleft license, which requires you to open the derived source code.
☆	V1048	CWE-1164		The 'out[0]' variable was assigned the same value.
☆	V760			Two identical blocks of text were found. The second block begins from line 472.
☆	V522	CWE-690	CERT-MEM52-CPP	There might be dereferencing of a potential null pointer 'bas'.
☆	V557	CWE-125	CERT-ARR30-C	Array overrun is possible. The value of 'pos' index could reach 2.
☆	V547	CWE-571		Expression 'uptr != NULL' is always true.

First time analysis



First time analysis

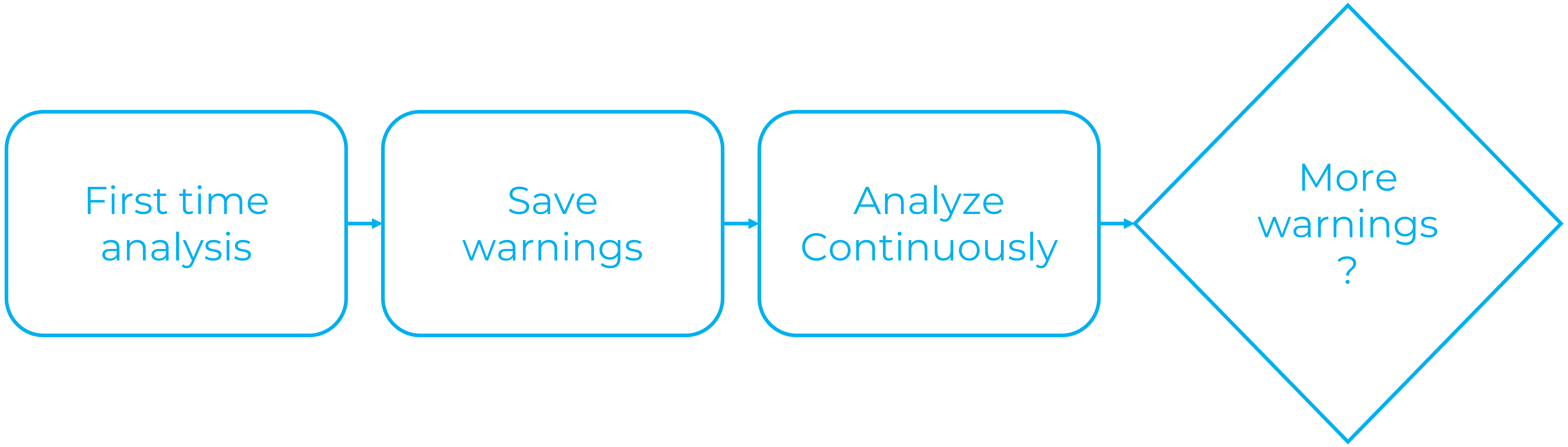
Save warnings

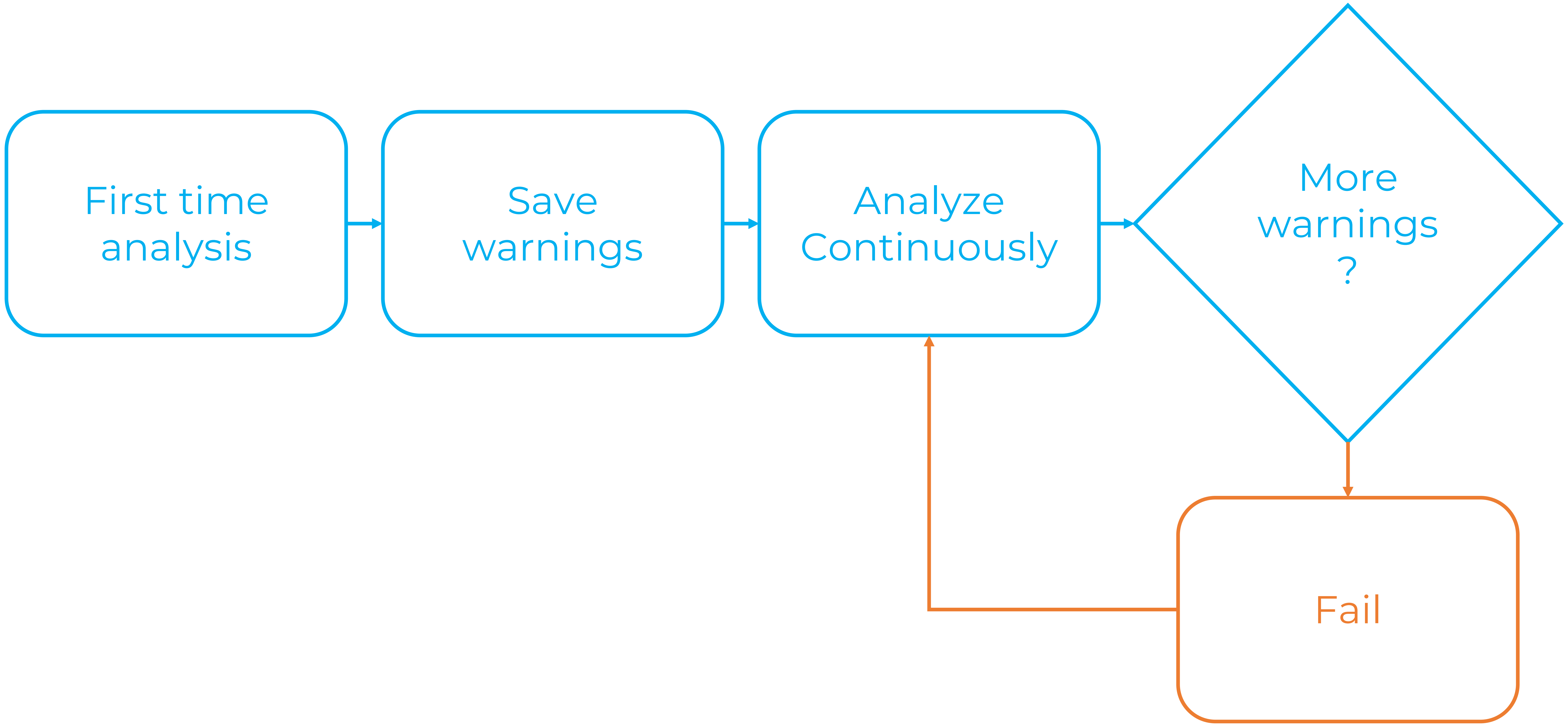


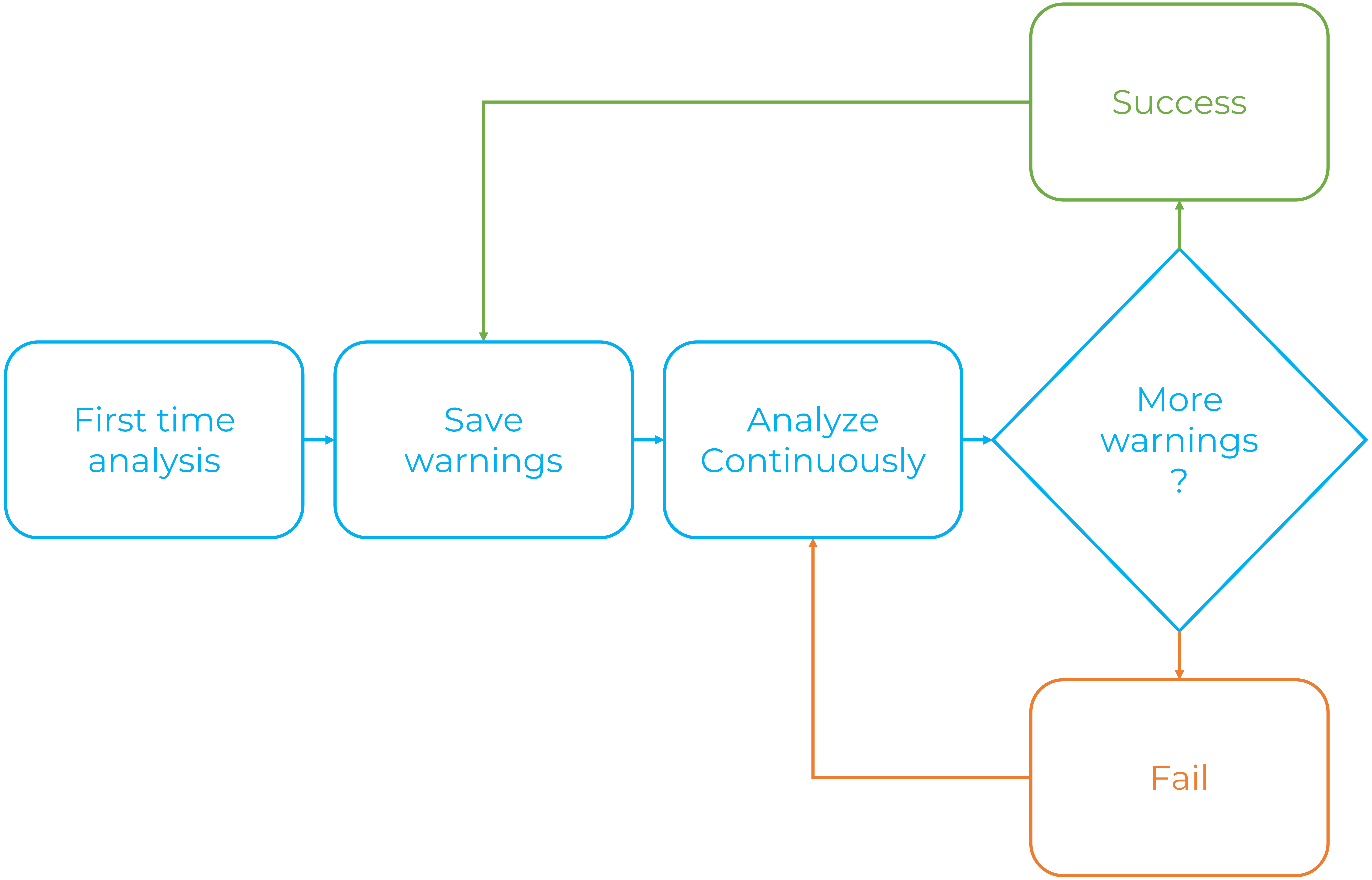
First time analysis

Save warnings

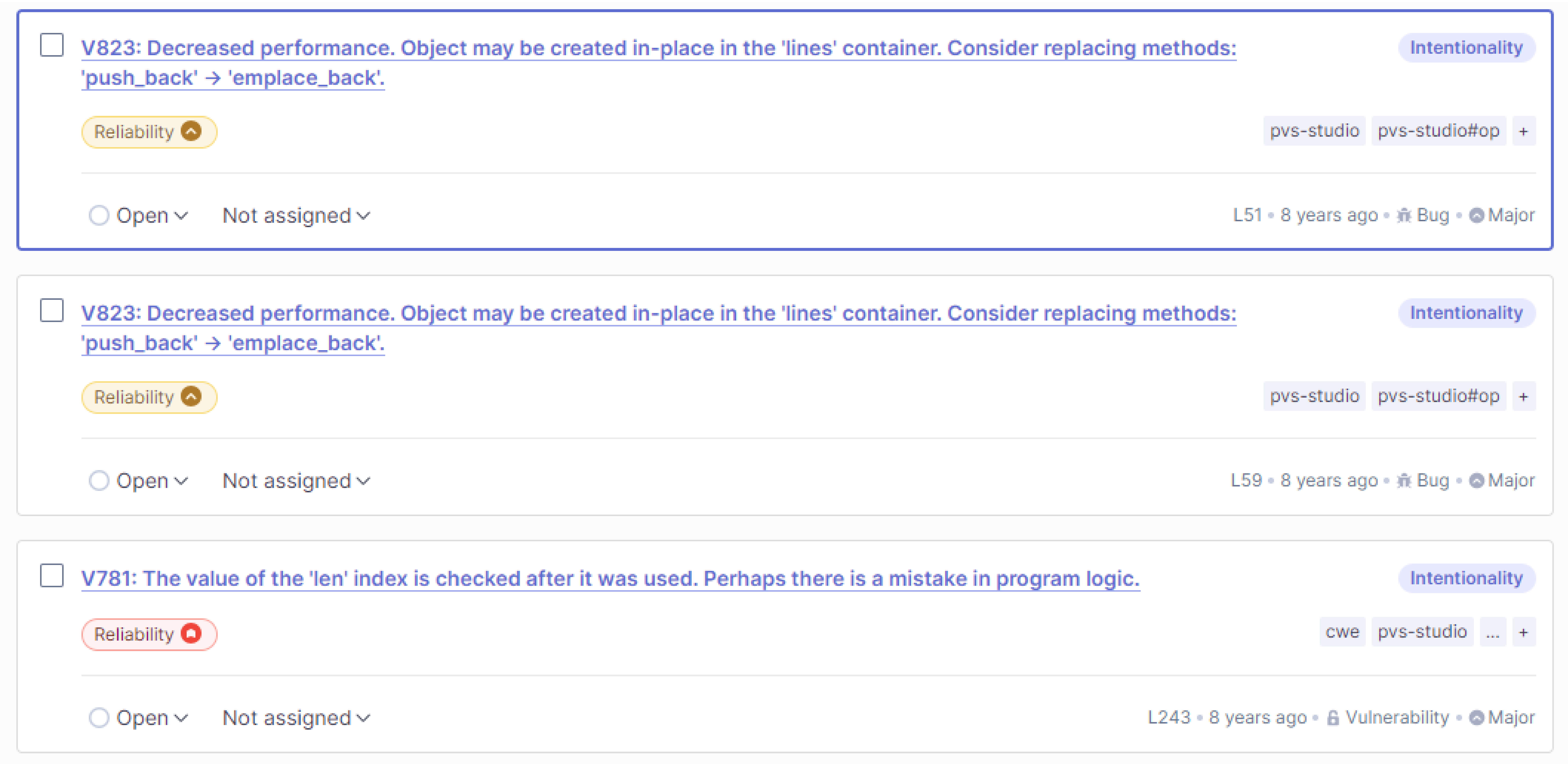
Analyze Continuously


First time analysis
Save warnings
Analyze Continuously
More warnings ?


First time analysis
Save warnings
Analyze Continuously
More warnings ?
Fail


Success
First time analysis
Save warnings
Analyze Continuously
More warnings ?
Fail

SonarQube



□ V823: Decreased performance. Object may be created in-place in the 'lines' container. Consider replacing methods: 'push_back' → 'emplace_back'.

Intentionality

Reliability

pvs-studio pvs-studio#op +

○ Open ⌄ Not assigned ⌄

L51 • 8 years ago • Bug • Major

□ V823: Decreased performance. Object may be created in-place in the 'lines' container. Consider replacing methods: 'push_back' → 'emplace_back'.

Intentionality

Reliability

pvs-studio pvs-studio#op +

○ Open ⌄ Not assigned ⌄

L59 • 8 years ago • Bug • Major

□ V781: The value of the 'len' index is checked after it was used. Perhaps there is a mistake in program logic.

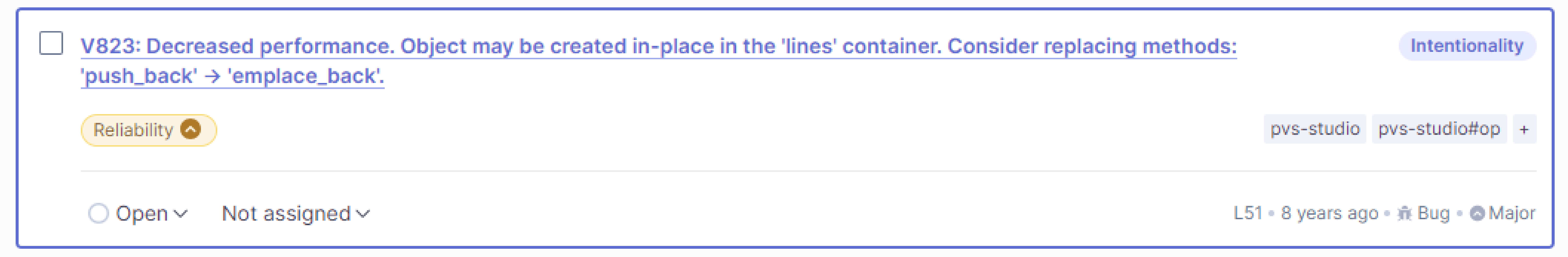
Intentionality

Reliability

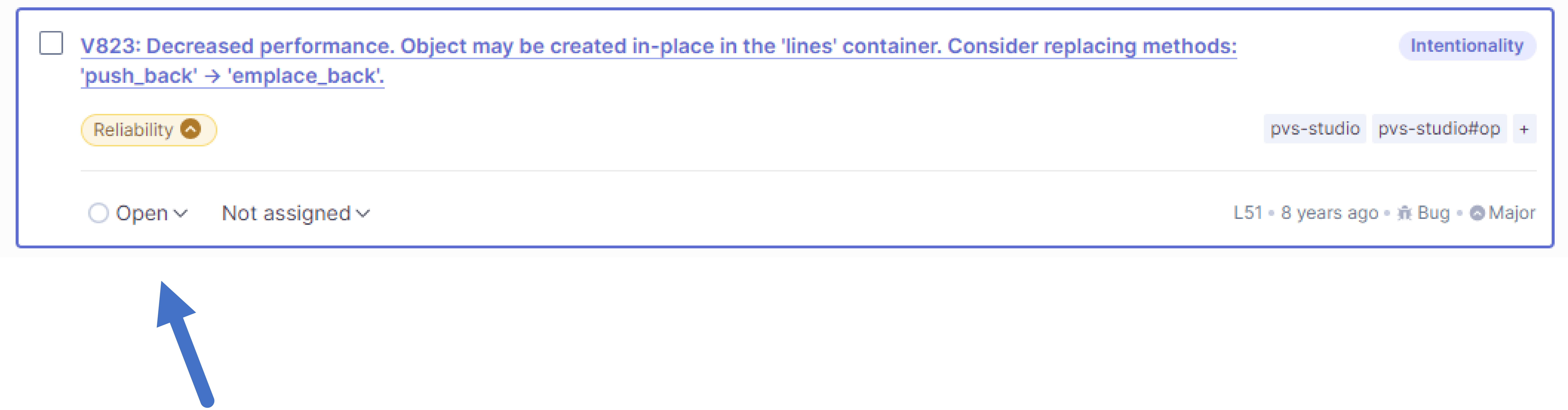
cwe pvs-studio ... +

○ Open ⌄ Not assigned ⌄

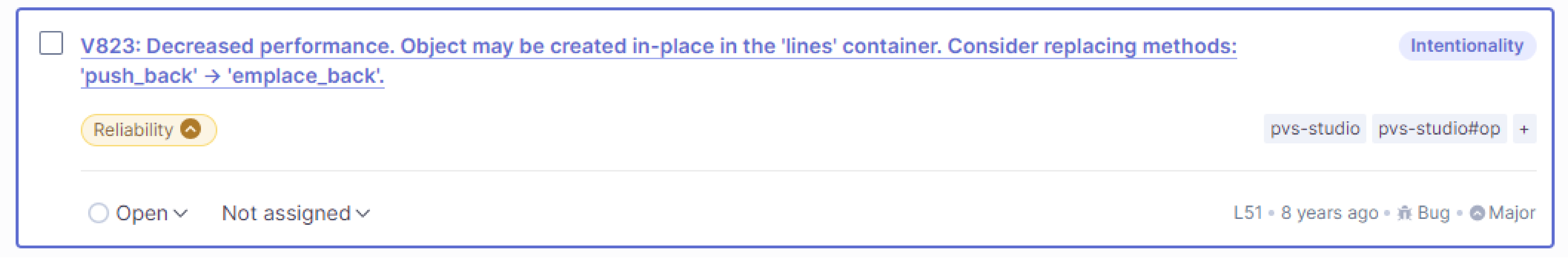
L243 • 8 years ago • Vulnerability • Major



□ V823: Decreased performance. Object may be created in-place in the 'lines' container. Consider replacing methods: 'push_back' → 'emplace_back'.



□ V823: Decreased performance. Object may be created in-place in the 'lines' container. Consider replacing methods: 'push_back' → 'emplace_back'.



□ V823: Decreased performance. Object may be created in-place in the 'lines' container. Consider replacing methods: 'push_back' → 'emplace_back'.

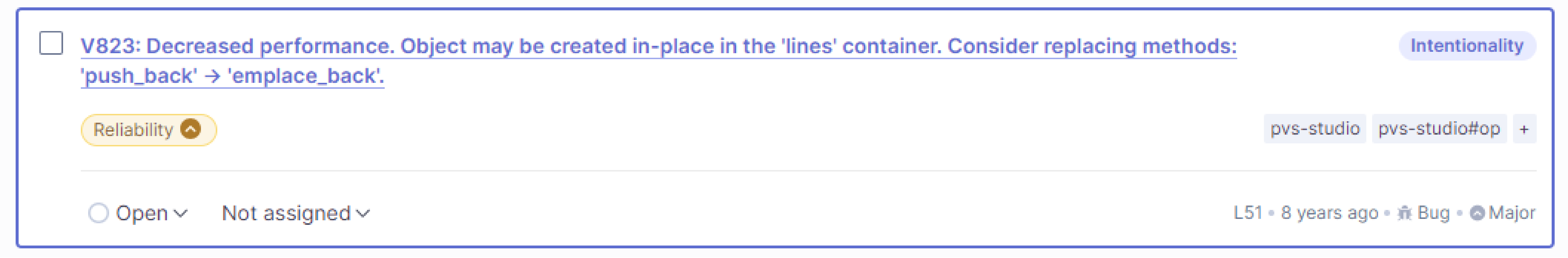
Intentionality

Reliability

pvs-studio pvs-studio#op +

○ Open Not assigned

L51 • 8 years ago • Bug • Major



□ V823: Decreased performance. Object may be created in-place in the 'lines' container. Consider replacing methods: 'push_back' → 'emplace_back'.

Intentionality

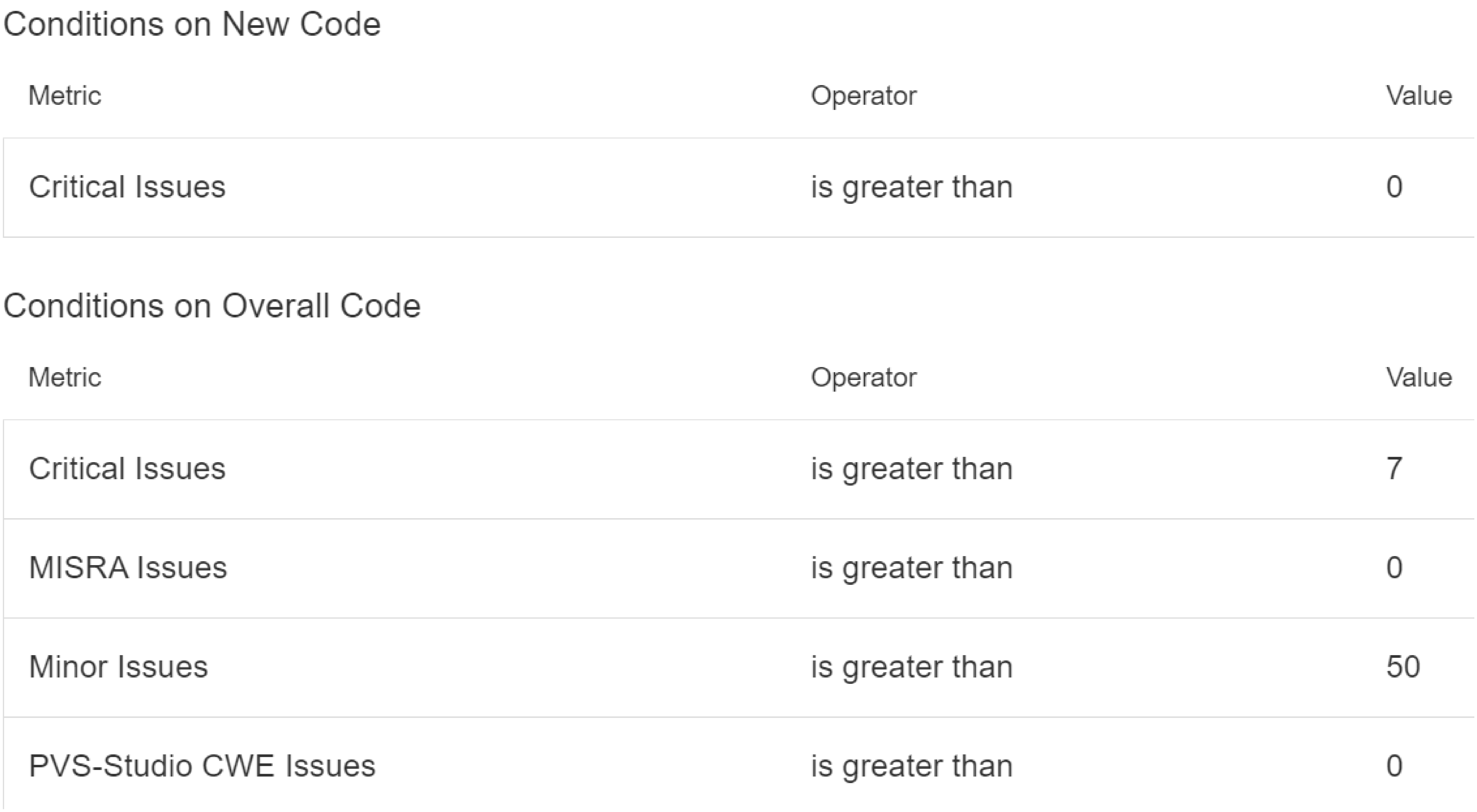
Reliability

pvs-studio pvs-studio#op +

○ Open

Not assigned

L51 • 8 years ago • Bug • Major

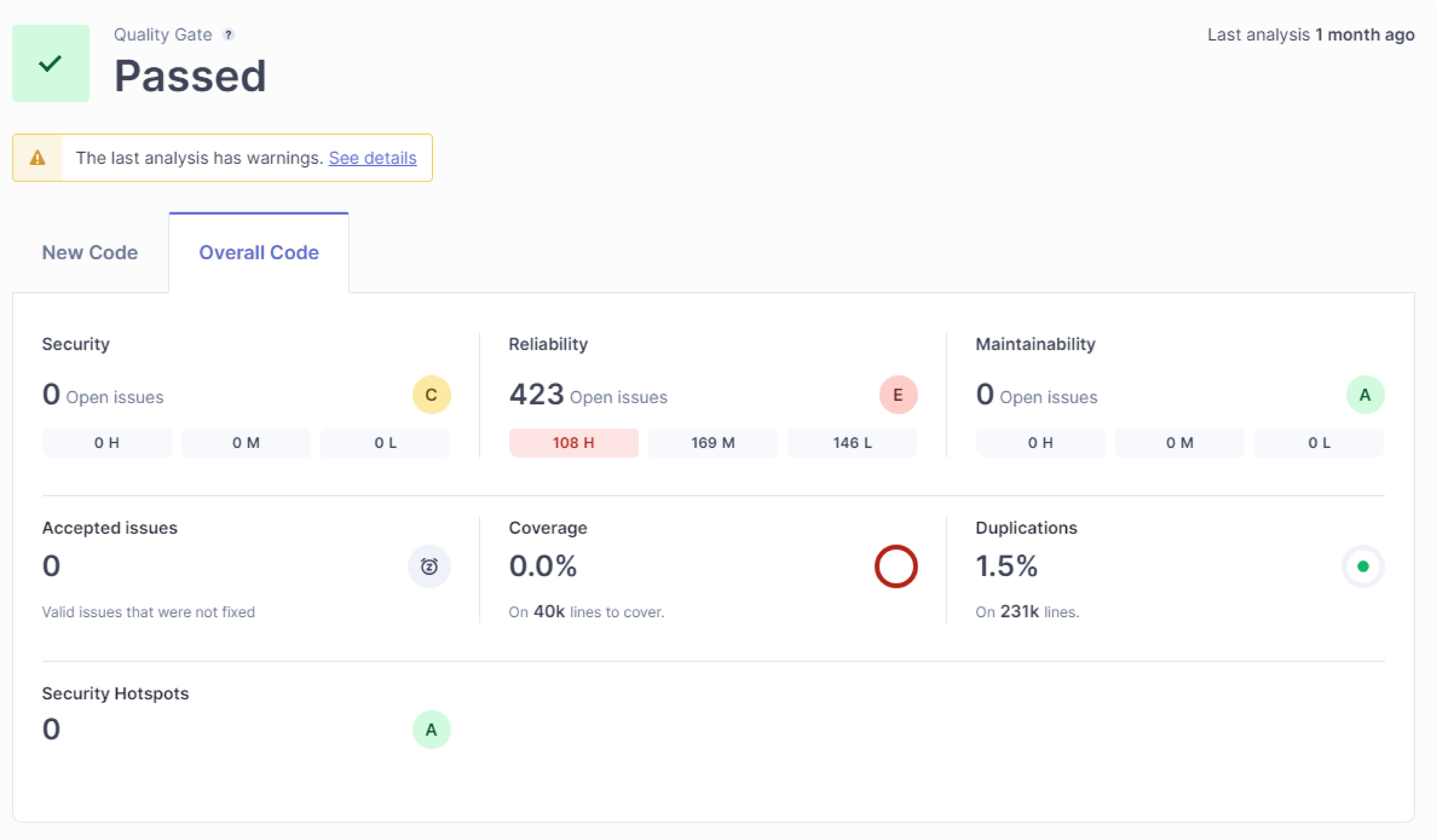


Conditions on New Code

Metric	Operator	Value
Critical Issues	is greater than	0

Conditions on Overall Code

Metric	Operator	Value
Critical Issues	is greater than	7
MISRA Issues	is greater than	0
Minor Issues	is greater than	50
PVS-Studio CWE Issues	is greater than	0



Quality Gate ?

Passed

Last analysis **1 month ago**

The last analysis has warnings. See details

New Code | Overall Code

Security

0 Open issues C

0 H | 0 M | 0 L

Reliability

423 Open issues E

108 H | 169 M | 146 L

Maintainability

0 Open issues A

0 H | 0 M | 0 L

Accepted issues

0

Valid issues that were not fixed

Coverage

0.0%

On **40k** lines to cover.

Duplications

1.5%

On **231k** lines.

Security Hotspots

0 A

```
curl --request POST \
   <address>/api/qualitygates/update_condition \
    -d ‘id=<quality_gate_id>' \
    -d 'error=7' \
    -d 'metric=critical_violations' \
    -d 'op=GT’
```

Conclusions

Conclusions

Conclusions

- SAST helps detect vulnerabilities early

Conclusions

- SAST helps detect vulnerabilities early
- Early detection significantly reduces the cost of fixing

Conclusions

- SAST helps detect vulnerabilities early
- Early detection significantly reduces the cost of fixing
- SAST in CI/CD makes security checks continuous

Q/A

Integrating SAST into DevSecOps

Click me!

Anton Tretyakov

C++ engineer

tretyakov@viva64.com

PVS-Studio