## Première exécution de Frama-C

Message d’erreur

Une image contenant texte, reçu, Police, blanc

Le contenu généré par l’IA peut être incorrect.

Lorsque qu’on execute la commande frama-c -eva -main is\_negative cert\_exp\_33.c, Frama-C analyse le code fourni en utilisant la fonction is\_negative comme point d'entrée. Durant cette analyse, Frama-C émet des messages d'avertissement concernant l'utilisation de la variable non initialisée sign. Ces avertissements signalent que la variable sign est lue sans avoir été initialisée, ce qui peut entraîner un comportement indéfini.​

Ce problème correspond à la règle **EXP33-C** du CERT C, intitulée "Do not read uninitialized memory". Cette règle stipule qu'il est essentiel de ne pas lire de la mémoire non initialisée, car cela peut conduire à des comportements imprévisibles et potentiellement dangereux.

En lançant la commande frama-c -eva -main is\_negative cert\_exp\_33.c on a un message d’erreur indiquant un comportement non définit. En ajoutant #include <stddef.h> au début du code, cela règle le problème.

Après avoir ajouté Frama\_C\_show\_each\_set\_flag(number, \*sign\_flag) et Frama\_C\_show\_each\_is\_negative(number, sign); on a le message suivant en éxecutant à nouveau :

Une image contenant texte, capture d’écran, Police, noir et blanc

Le contenu généré par l’IA peut être incorrect.

Avec n= 1

Une image contenant texte, capture d’écran, Police, document

Le contenu généré par l’IA peut être incorrect.

Pour n =2

Une image contenant texte, capture d’écran, Police, document

Le contenu généré par l’IA peut être incorrect.

**Le partitionnement améliore la précision de l’analyse** en séparant bien les différents cas (nombre négatif, positif ou nul).

En remplaçant if (number>0) par if (number>=0) on traite le cas number=0 qui n’était pas traité et qui était à l’origine du problème

Nous n’avons plus d’alarme :

Une image contenant texte, capture d’écran, Police, reçu

Le contenu généré par l’IA peut être incorrect.

## CERT C EXP33-C bis

Ici la fonction qui explore le plus de possibilités possible est la fonction func()

Nous allons donc l’utiliser comme point d’entrée.

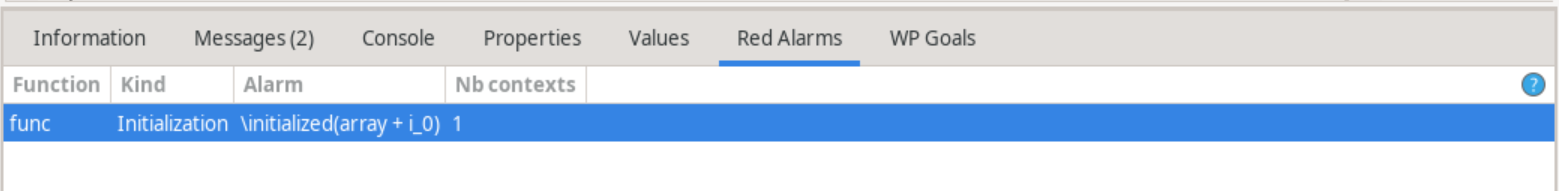
Une image contenant texte, capture d’écran, Police, noir et blanc

Le contenu généré par l’IA peut être incorrect.

L’alarme correspondant à la lecture d’une location mémoire non-initialisée est la première alarme : 1 access to unintialized left-values.

On obtient une Red Alarm avec la commande :

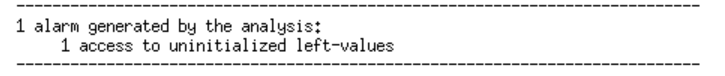
frama-c-gui -eva -eva-min-loop-unroll 10 -main func cert\_exp\_33\_realloc.c



En ajoutant l’annotation Une image contenant texte, capture d’écran, Police, ligne

Le contenu généré par l’IA peut être incorrect.

Split ret ==0, nous n’avons plus qu’un message d’erreur :

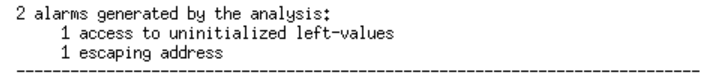


En utilisant la version corrigée proposée par CERT, on obtient de nouvelles erreurs :

Une image contenant texte, capture d’écran, Police, blanc

Le contenu généré par l’IA peut être incorrect.

J’ai réussi à réduire ce nombre d’erreurs à 2 en utilisant des return au niveau des /\* Handle error\*/ du code corrigé :



CWE 20

En lançant Frama-C/Eva sur le code on obtient 8 erreurs :

Une image contenant texte, reçu, Police, capture d’écran

Le contenu généré par l’IA peut être incorrect.

Les erreurs integer overflows sont du à la ligne :

If m \* n exceeds INT\_MAX / sizeof(board\_square\_t qui cause un **integer overflow**, menant à une mauvaise allocation en mémoire (il s’agit d’une vulnérabilité CWE-190)

L’erreur Unitialized left-values est du au fait que board peut être NULL, ce qui conduit à un comportement non défini, notamment dans la ligne :

if (board[(m-1)\*n] == 0) return 0;

Le problème de invalid memory access peut être du à board[(m-1)\*n], où dans le cas où m=0 on a un indice négatif.

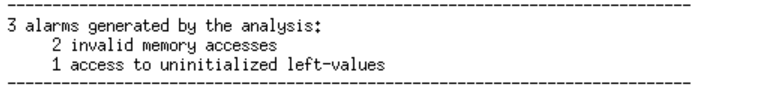
Patch1 :

En vérifiant les dimensions de m et n on élimine les problèmes d’integer overflow avec le code suivant

Une image contenant texte, Police, capture d’écran, ligne

Le contenu généré par l’IA peut être incorrect.

On obtient ainsi :



Patch 2

On vérifie si malloc a échoué avec le code suivant :

Une image contenant texte, Police, capture d’écran, ligne

Le contenu généré par l’IA peut être incorrect.

Cependant, on obtient toujours les mêmes alarmes.