STATIC CODE ANALYSIS

Sistemes per analisis estàtic de codi per a detectar més errors?

L'anàlisi de codi estàtic és un mètode de depuració, mitjançant l'estudi del codi font abans d'executar un programa. Es tracta d'intentar veure si el codi en concret compleix una sèrie de regles de codificació. Aquest tipus d'anàlisi aborda les debilitats del codi font que poden provocar vulnerabilitats. Per descomptat, això també es pot aconseguir mitjançant revisions manuals de codi. Però utilitzar eines automatitzades és molt més efectiu.

STATIC VS DYNAMIC

Aleshores, quina diferència hi ha entre l'anàlisi estàtic i l'anàlisi dinàmic?

Tots dos tipus detecten defectes. La gran diferència és on trobem defectes en el cicle de vida del desenvolupament.

L'anàlisi estàtica identifica els defectes abans d'executar un programa.

L'anàlisi de codi dinàmic identifica els defectes després d'executar un programa. Tanmateix, és possible que alguns errors de codificació no apareguin durant les proves, però que tot i aix, no s'obtingui el resultat esperat. Per això és important realitzar les proves estàtiques, ja que ens permeten detectar l'error.

MÈTODES UTILITZATS PER LES PROVES ESTÀTIQUES

- Mètodes formals: els resultats d'aquest tipus de proves provenen exclusivament de l'aplicació d'algorismes matemàtics. Aquests utilitzen semàntica, semàntica denotacional, semàntica axiomàtica, semàntica operativa i la interpretació bàsica.

- Interpretació abstracta: aquest crea un sistema abstracte que imita a l'original. Aquesta imita el comportament del codi original, però al ser abstracte, es fa més senzill d'analitzar tot i que algunes propietats del sistema abstracte poden no ser certes en el sistema real a causa de la inexactitud del sistema abstracte.

- Anàlisi de dades: tècnica basada en la recopilació d'informació sobre el possible conjunt de valors.

- Llògica Hoare: és un sistema formal que, utilitzant un conjunt de regles llogiques decideix la correcció dels programes.

- Verificació de models: aquest considera que els sistemes tenen un estat finit o que es poden arribar a reduir a un model d'abstracció.

- Execució simbòlica: s'utilitza per a derivar expresions matemàtiques que representen el valor de variables que canvien en punts determinats del codi.

ADRESS SANITIZERS

AddressSanitizer és una eina de programació de codi obert que detecta errors de corrupció de memòria, com ara desbordaments de buffer o accessos a un punter penjat.

Les eines més populars que ens permeten dur a terme aquest anàlisi estàtic per als llenguatges de la familia de C són:

- GCC

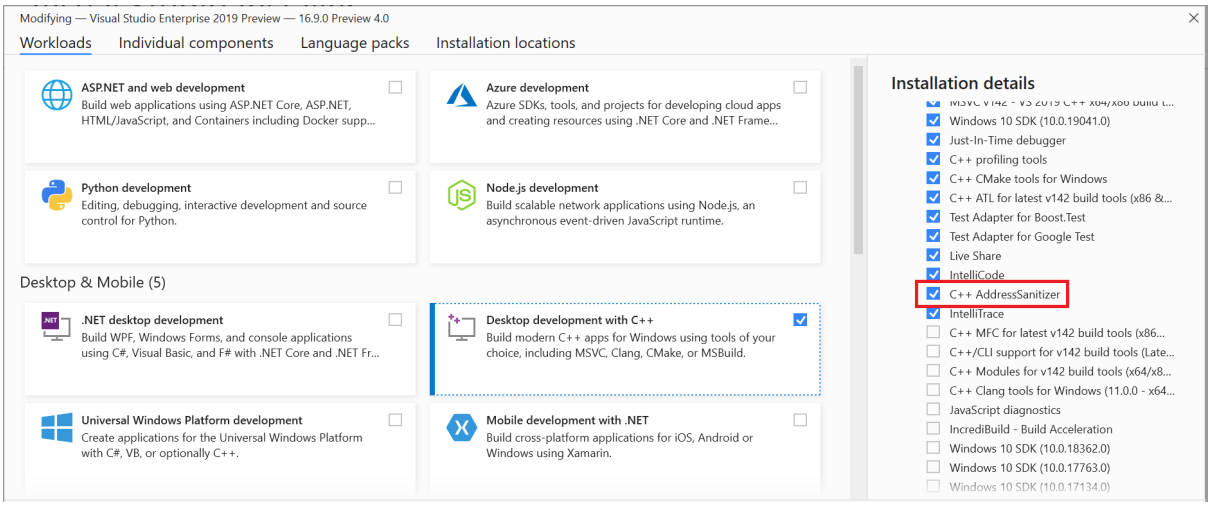
És un conjunt de compiladors per a diferents llenguatges de programació que implementa mètodes d'adress sanitizers per tal de generar programes binaris sense errors. Aquest no porta implementat l'analitzador estàtic, però si s'afegeixen paràmetres en la seva comanda es pot activar l'analitzador.

- Clang

Clang és un compilador per a llenguatges de programació de la familia C que intenta remplaçar als compiladors de GCC, ja que presenta serveix similars. A diferencia de GCC, Clang porta incorporat l'analitzador directament, de tal forma que, abans de compilar un codi C, aquest busca errors del codi fent servir les pràctiques d'anàlisi estàtic.

- Visual studio

Visual studio és un IDE que pot integrar AdressSanitazers, ja que el compilador amb el que es compilen els programes s'escull. Per tant si s'escull un compilador amb AdressSanitizers podem compilar el nostre codi utilitzant aquestes pràctiques directament.



EXEMPLES D'EINES ONLINE PER A L'ANÀLISI DE CODI ESTÀTIC

Embold, Collaborator (Softbear), Coverity

EXEMPLES D'EINES OFFLINE PER A L'ANÀLISI DE CODI ESTÀTIC

Sonarqube, PVS studio, CodeScene, cppcheck