# Progetto CodeStresser

CodeStresser prende semplicemente del codice e ti dice se c’è qualche vulnerabilità nota.

## Organizzazione del codice

Il codice verrà organizzato in moduli.

|  |  |
| --- | --- |
| Nome | Descrizione |
| Lettore del Codice | Prende in input il codice da analizzare e lo carica nel programma |
| Parser | Fai il parsing del codice in AST (Abstract Syntax Tree) |
| Analizzatore Statico | Raccoglie tutti i punti di possibile vulnerabilità nel codice |
| Modello di ML | Deve essere in grado di predire la probabilità di exploiting delle vulnerabilità trovate dall’analizzatore statico e categorizzare quindi i punti di possibile vulnerabilità |
| Generatore del report | Genera il report |

## Dettagli generici

L’applicazione verrà sviluppata in python per ovvi motivi legati all’uso di ML.

## Moduli

### Lettore del codice

Il lettore del codice consisterà in un semplice modulo che prenderà in input uno o più file di codice per inserirli nel programma (lista di file).

### Parser

Utilizza l’output del lettore del codice per generare un Abstract Syntax Tree.  
Un **Abstract Syntax Tree (AST)** è una rappresentazione strutturata e gerarchica del **significato sintattico di un codice sorgente**, senza includere dettagli inutili della sintassi (come parentesi o punteggiatura).

A screenshot of a computer

AI-generated content may be incorrect.

Il modulo utilizzerà la libreria “ast” di python per la trasformazione in Abstract Syntax Tree.

A screenshot of a computer program

AI-generated content may be incorrect.