<u>Rapport exercice 2</u> : Test de vulnérabilités avec semgrep

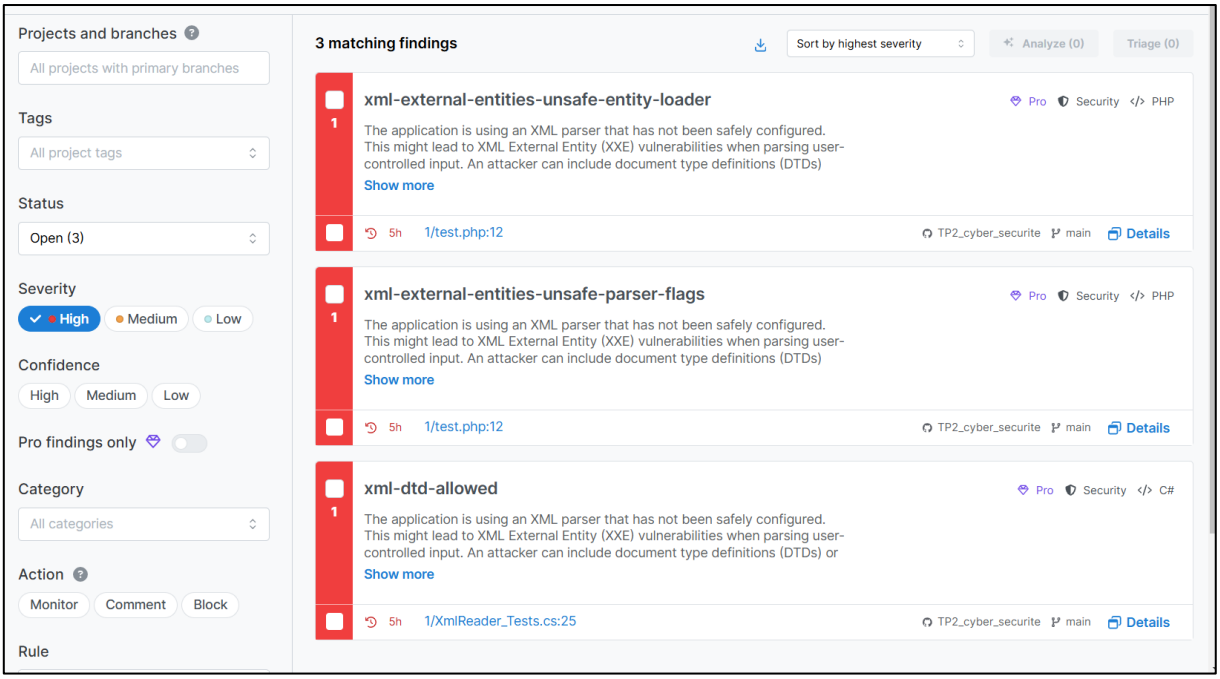
Etape 1 : Intégration du dépôt git à semgrep

J'ai créé un dépôt git content tous les fichiers liés à l'application. Ensuite, j'ai lié le dépôt à semgrep cloud ce qui permet de :

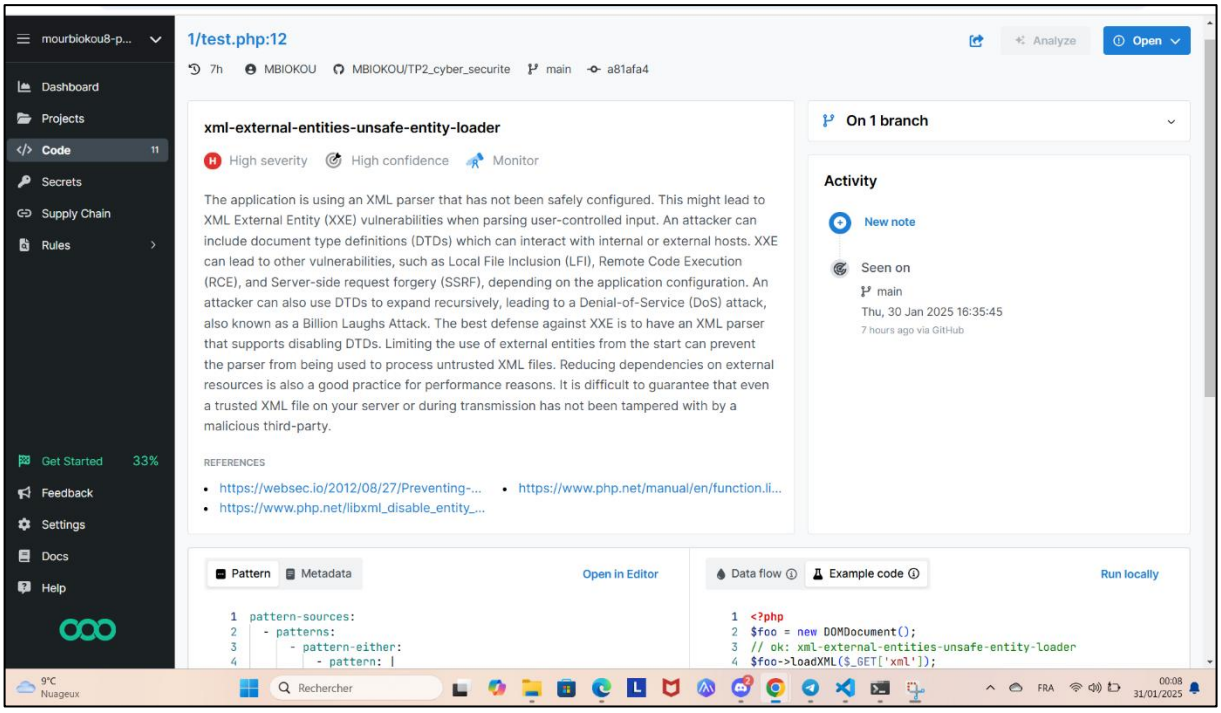
- Analyser automatiquement votre code source.
- Utiliser un ensemble de règles prédéfinies (OWASP Top 10, Best Practices DevSecOps).
- Générer un rapport détaillant les vulnérabilités identifiées (avec des explications pour chaque problème).

Etape 2 : Analyse des vulnérabilités

Après avoir lié mon dépôt git à semgrep, j'ai lancé le scan. Il analyse le dépôt en ligne, génère des alertes catégorisées par High, Medium, et Low, et propose des recommandations de correction. Voici ce que présente semgrep après le scan :



On remarque la présence de 2 vulnérabilités ***higths*** dont l'une dans le fichier ***test.php*** et l'autre dans le fichier ***XmlReader_Test.cs***. Plus en détail :



On remarque qu'il s'agit d'une vulnérabilité liée à l'XML External Entity (XXE). L'injection XXE est un type d'attaque contre une application qui analyse les entrées XML. Une telle vulnérabilité peut entrainer une fuite de données, une exécution de code à distance (RCE), Denial of Service (DoS). Pour la corriger il faut donc désactiver les DTDs dans le parser XML et limiter les entités externes :

```php
// Désactiver le chargement d'entités externes
libxml_disable_entity_loader(true);

$xml = strlen($_GET['xml']) > 0 ? $_GET['xml'] : '<root><content>No XML fo

$document = new DOMDocument();
//1
$document->xmlStandalone = true;

$document-> localXML($xml,LIBXML_NOENT);
```

Une fois les vulnérabilités corrigées, j'ai relancé le scan, consulté les sections High, Medium, et Low :