

OUTIL D'AUDIT DE CODE

Yasca

Ce guide décrit l'installation et l'utilisation de l'outil d'audit de code source « yasca » sur un système windows XP/Vista/7. Yasca est un analyseur de code source qui intègre d'autres outils open-source (PMD, FindBugs, jlint) afin de générer un rapport.





الوكالة الوطنية للأمن المعلوماتية
Agence Nationale de la Sécurité Informatique

Gestion des versions du document

Version	Date	Modification apportée
1.0	27/08/2010	Première version

PLAN

1.	Presentation.....	3
2.	Fonctionnement.....	4
3.	Installation	4
3.1.	Prérequis.....	4
3.2.	Yasca.....	4
3.3.	Plugins.....	5
3.4.	Test de fonctionnement.....	5
4.	Utilisation.....	6
❖	Generation du rapport.....	6
❖	Exemples de scénarios	6

1. PRESENTATION

Yasca est un outil open source qui fait le scan des failles de sécurité, de la qualité, de la performance et de la conformité aux meilleures pratiques dans le code source d'un programme.

Il s'agit d'un outil en ligne de commande qui génère des rapports au format HTML, CSV, XML, SQLite et d'autres formats.

Yasca a au moins un scanner pour chacun des types de fichiers suivants:

- Java
- C/C++
- .NET (VB.NET, C#, ASP.NET)
- PHP
- ColdFusion
- COBOL
- HTML
- JavaScript
- CSS
- Visual Basic
- ASP
- Python
- Perl

Afin de scanner certains types de fichiers, Yasca intègre d'autres plugins open source, notamment:

- FindBugs
- PMD
- Jlint
- JavaScript Lint
- PHPLint
- Cppcheck
- ClamAV
- Rats

- Pixy
- Fxcop
- FindBugs-plugins

Ces plugins peuvent être téléchargeables à partir de ce lien :

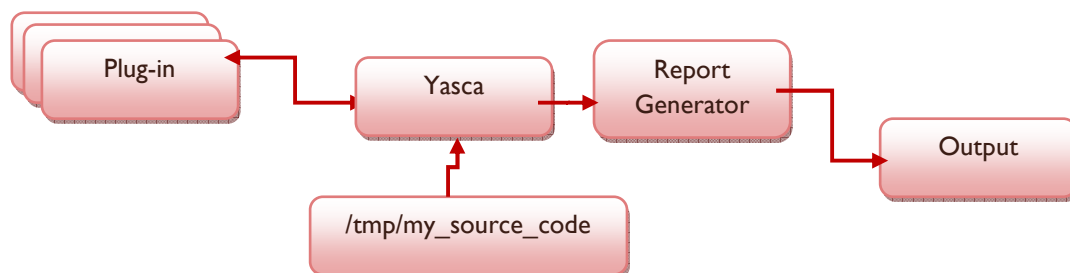
<http://sourceforge.net/projects/yasca/files/>

2. FONCTIONNEMENT

➤ Architecture basée sur des plug-ins :

- ✚ Major Plug-in : yasca fait appel à un autre outil pour faire le scan. Par exemple PMD, J-Lint, FindBugs, Pixy, Grep
- ✚ Minor Plug-in : yasca utilise la logique embarqué. Relativement facile à écrire, extrêmement flexible

➤ Schéma de fonctionnement :



3. INSTALLATION

3.1. Prérequis

- Java JRE (ou JDK) v1.5 or later (prérequis pour les plugins PMD et FindBugs)
- 512 MB RAM au minimum.

3.2. Yasca

- Téléchargez Yasca depuis ce lien, puis décompressez-la dans un répertoire :

Par exemple : « `c:\yasca` »

<http://sourceforge.net/projects/yasca/files/Yasca%202.x/Yasca%202.1/yasca-2.1.zip/download>

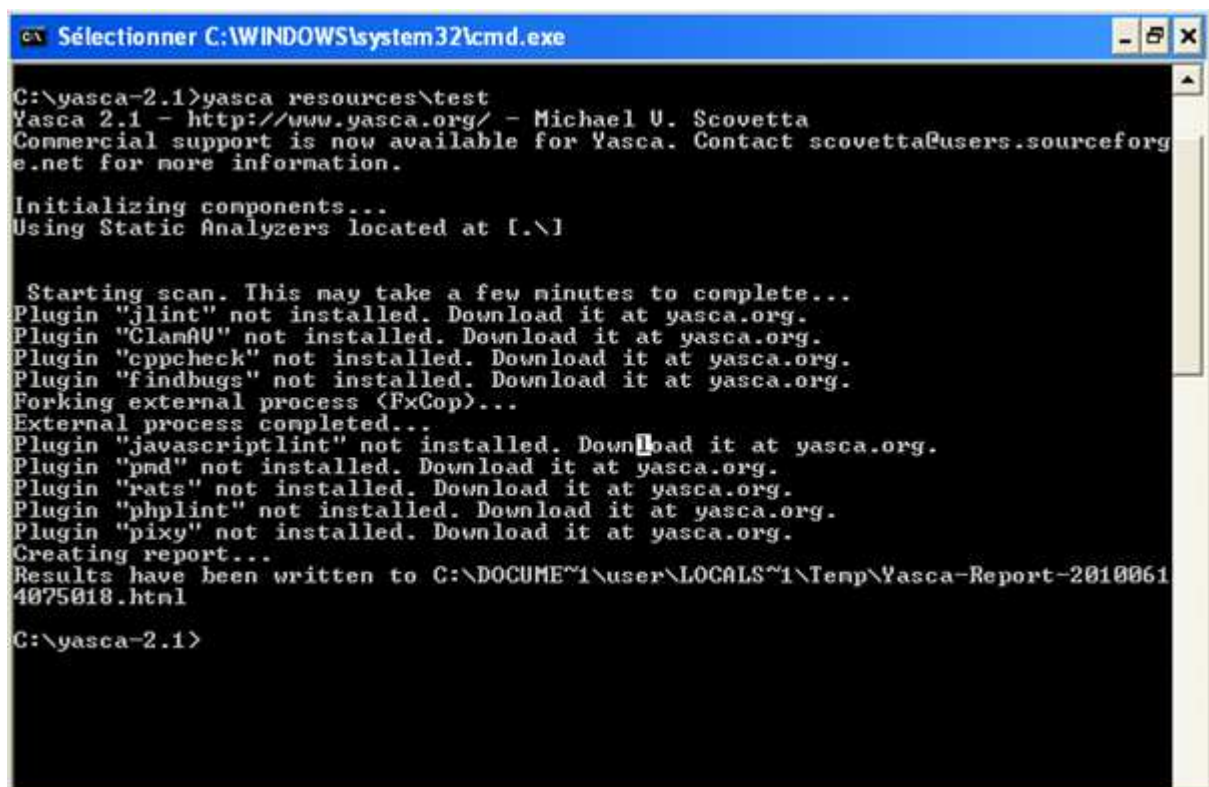
3.3. Plugins

- Notez que pour certains plugins, vous devez avoir Java installé comme pré requis.
- Si vous souhaitez utiliser des plug-ins tiers, téléchargez-les depuis ce lien : <http://sourceforge.net/projects/yasca/files/>
- Installez-les dans le même répertoire que Yasca. Par exemple, vous pouvez installer le plugin PMD en: « *c:\yasca* »

3.4. Test de fonctionnement

- A partir du répertoire d'installation Yasca, scannez «resources\test».

```
c:\yasca>yasca resources\test
```



```
C:\yasca-2.1>yasca resources\test
Yasca 2.1 - http://www.yasca.org/ - Michael U. Scovetta
Commercial support is now available for Yasca. Contact scovetta@users.sourceforge.net for more information.

Initializing components...
Using Static Analyzers located at [.\]

Starting scan. This may take a few minutes to complete...
Plugin "jlint" not installed. Download it at yasca.org.
Plugin "ClamAV" not installed. Download it at yasca.org.
Plugin "cppcheck" not installed. Download it at yasca.org.
Plugin "findbugs" not installed. Download it at yasca.org.
Forking external process (FxCop)...
External process completed...
Plugin "javascriptlint" not installed. Download it at yasca.org.
Plugin "pmd" not installed. Download it at yasca.org.
Plugin "rats" not installed. Download it at yasca.org.
Plugin "phplint" not installed. Download it at yasca.org.
Plugin "pixy" not installed. Download it at yasca.org.
Creating report...
Results have been written to C:\DOCUME~1\user\LOCALS~1\Temp\Yasca-Report-20100614075018.html
C:\yasca-2.1>
```

- Un rapport sera, par défaut, générer en html et sauvegarder dans un répertoire « yasca » sur le bureau de l'utilisateur.

4. UTILISATION

❖ Generation du rapport

Vous pouvez modifier l'emplacement et l'extension du rapport en utilisant l'option « -o nouveau_chemin » lors du lancement du scan.

- Si vous voulez générer un rapport.csv sous le même répertoire «yasca».

```
c:\yasca> yasca resources\test -o ./rapport.csv
```

- Vous pouvez créer une base de données SQLite contenant le rapport comme ci-dessous :
 - Créez une base de données MySQL, nommée yasca par exemple, via phpMyAdmin ou easyphp.
 - Exécutez le script « *etc/yasca.mysql* » pour créer la structure de la table nécessaire.
 - Utilisez l'option "-d" option de ligne de commande à passer les informations d'identification de base de données au moteur Yasca.

Par exemple :

```
c:\yasca> yasca -d « SQLReport.database=./my.db » -r SQLReport ./yasca/
```

❖ Exemples de scénarios

➤ *Scenario 1 : Projet PHP Simple*

- Supposons que vous avez le code source d'un projet PHP placé dans c:\projet\mon_projet_php\src. Vous pouvez donc l'auditer via cette commande :

```
c:\yasca> yasca c:\projet\mon_projet_php\src
```

- Dans ce cas, le rapport sera généré, par défaut, dans un répertoire nommé yasca sur le bureau.

- Tant que vous n'avez aucun fichier java, C, C++ dans votre projet, vous pouvez exclure quelques plugins qui n'ont rien à scanner. Voici la commande :

```
c:\yasca> yasca -px FindBugs,Antic,JLint,PMD c:\projet\mon_projet_php\src
```

➤ *Scenario 2 : Projet Java, Output dans un autre répertoire*

- Supposons que vous avez un projet Java placé dans c:\projet\mon_projet et vous voulez que le rapport d'output soit sauvegardé dans un répertoire partagé (Z:\Yasca_Output). Vous pouvez ainsi exécuter cette commande :

```
c:\yasca> yasca -o z:\Yasca_Output c:\projet\mon_projet
```

➤ *Scenario 3 : Voir seulement les questions critiques*

- Vous pouvez régler un rapport afin d'afficher uniquement les questions critiques en utilisant un niveau de paramètre:

```
c:\yasca> yasca --level 1 c:\projet\mon_projet
```

❖ Snapshot

