Contenus Suspects

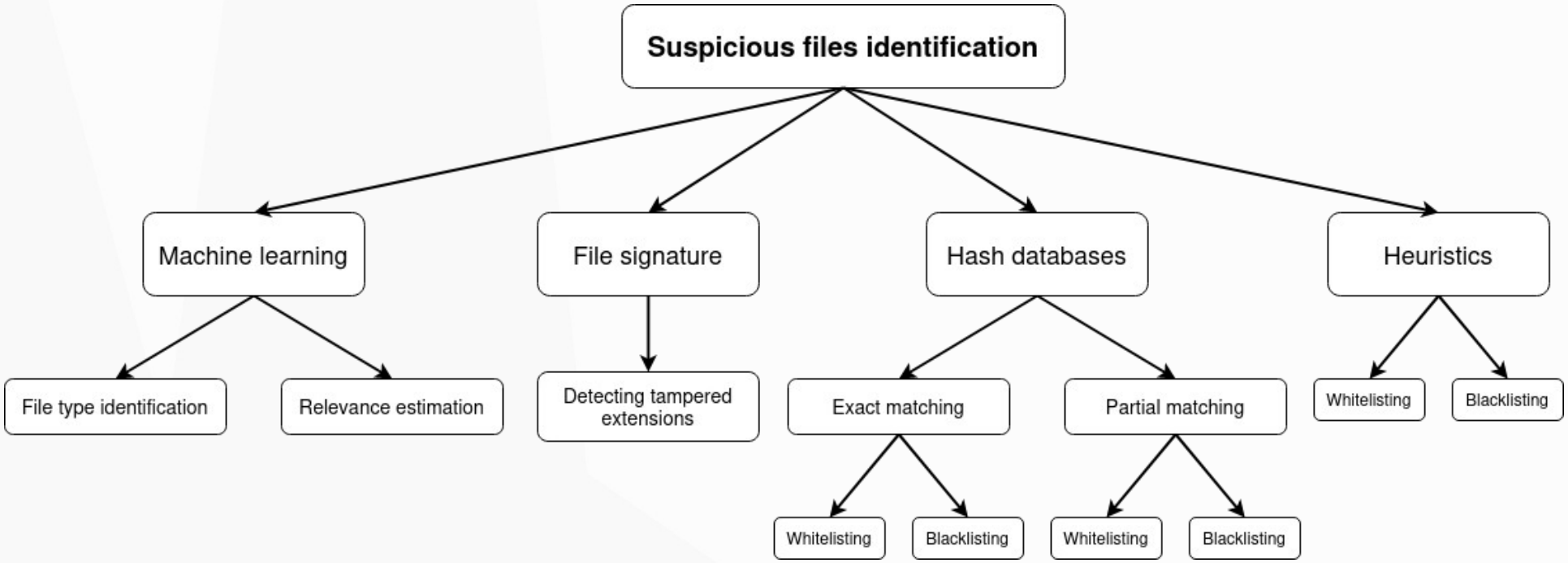
Matéo Delerue-Houard
Sylia Bouimedj

Objectifs :

- **Trouver des fichiers cachés par des utilisateurs sans connaissance techniques :**
  - **Modification des extensions de fichiers**
  - **Déplacement de fichiers dans des répertoires légitimes**
  - **Fichiers marqués comme cachés**

  Ne concerne pas :
  - **Chiffrement**
  - **Stéganographie**
  - **Anti-forensic**
- **Filtrage des fichiers légitimes**
- **Outil rapide**

État de l’art


Suspicious files identification
Machine learning
File signature
Hash databases
Heuristics
File type identification
Relevance estimation
Detecting tampered extensions
Exact matching
Partial matching
Whitelisting
Blacklisting
Whitelisting
Blacklisting
Whitelisting
Blacklisting

Implémentation

- **Programme écrit en C++**
- **Identification du type de ficher avec fidentify[1]**
- **Base de donnée de hash (schéma RDS)**
- **Recherche par mot clé**

[1] : *https://www.cgsecurity.org/wiki/PhotoRec*

Détection de fichiers cachés

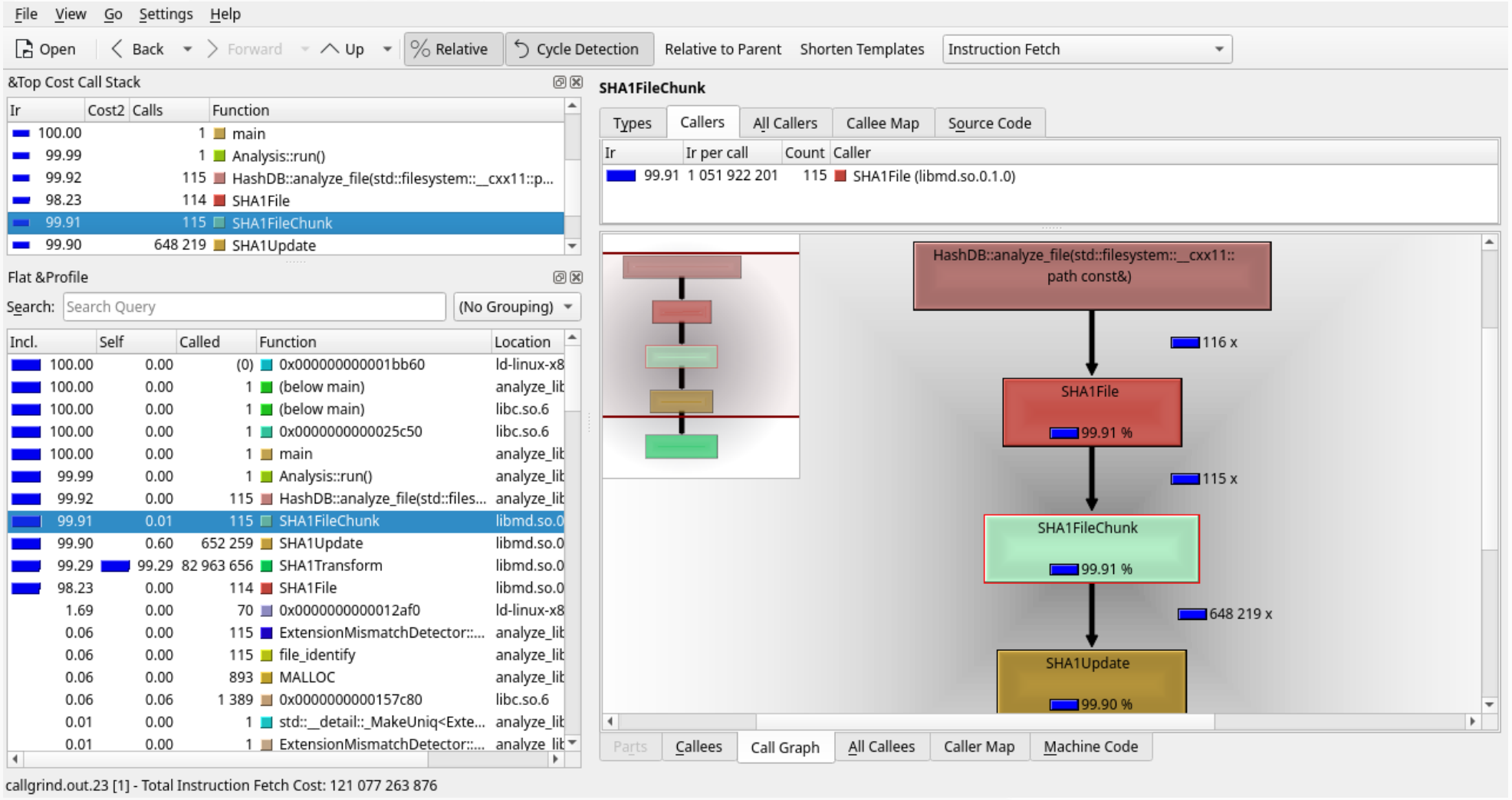
- **Fichiers cachés situés dans les répertoires utilisateurs (Vidéos, Images, Téléchargements…) considérés comme suspects**
- **Support de Windows & plateformes conforme au standard XDG (Linux, BSD…)**

Nouvelles heuristiques

- **Filtrage par taille : élimination des trop petit fichiers**
- **Recherche par intervalle de dates**

Optimisation

- **Profilage du code avec callgrind**



Optimisation

- **Benchmark OpenSSL vs libmd :**

```
Benchmark 1: ./analyze_libmd --timestamp 0 -d rds.sqlite3 -k keywords.txt root
  Time (mean ± σ):      3.469 s ±  0.121 s    [User: 1.761 s, System: 1.707 s]
  Range (min … max):    3.405 s …  3.808 s    10 runs

Benchmark 2: ./analyze_openssl --timestamp 0 -d rds.sqlite3 -k keywords.txt root
  Time (mean ± σ):      1.048 s ±  0.052 s    [User: 0.669 s, System: 0.378 s]
  Range (min … max):    1.010 s …  1.180 s    10 runs

Summary
  ./analyze_openssl --timestamp 0 -d rds.sqlite3 -k keywords.txt root ran
    3.31 ± 0.20 times faster than ./analyze_libmd --timestamp 0 -d rds.sqlite3 -
k keywords.txt root
```

Gains de performance : x3,3

Protocole d’évaluation

1. **Créer un volume de stockage légitime (installation d’OS)**
2. **Dissimulation de contenus :**
   - **Placement de fichiers dans des endroits aléatoires**
   - **Changement aléatoire des extensions**
   - **Marquer comme cachés des fichiers dans des dossiers utilisateurs**
3. **Tester chaque techniques individuellement et en combinaison**
4. **Relever leur taux de détection et de faux-positifs**

Résultats

Technique	Taux de détection	Faux-positifs
Identification du type de fichier	73 %	1099
Base de donnée de hash	35 %	11
Détection de fichier caché	3 %	0
Recherche par date	100 %	0
Tous combinés	100 %	1108

Pistes d’amélioration

- **Précision :**
  - **Recherche par types de fichiers (réduction des faux-positifs)**
  - **Ré-architecturer la base de donnée en arbre de Merkle**

    → Brise la compatibilité avec le RDS

- **Performances :**
  - **Implémentations plus performantes (DBMS, meilleurs algorithmes…)**
  - **Mappage mémoire des fichiers volumineux (mmap)**
  - **Parallélisation**