# Introduction au reverse Engineering

UNIVERSITÉ PARIS-EST
MARNE-LA-VALLÉE

Esipe
INGÉNIEURS PARIS EST

Quentin Bernard

IR3

# Sommaire

- Introduction
- Concepts
- Outils et Techniques utilisées
- Démo

# Introduction - Souvenez vous ….

- Cas concret :
  - Vous venez de finir votre projet java
  - Vous envoyez votre projet contenant les binaires à votre professeur
  - Le soir de l'envoi, dépité, vous décidez de supprimer vos sources
  - Au second semestre, un portage sur android vous est demandé ….
  - Mais que faire?

Du Reverse Engineering !!!

# Introduction - Définition

**Reverse Engineering**: Etude et analyse d’un système pour en déduire son fonctionnement interne

# Introduction - Contexte

- Pourquoi faire du reverse engineering?
  - Sécurité informatique
  - Améliorer la qualité des logiciels
  - Reconstitution du système original (Samba, pilote nouveau, …)

# Introduction - Compétences

- Compétences pour faire du reverse?
- Savoir développer!
- Connaître les spécifications du processeur cible (CISC : x86, RISC : ARM, …)
- Avoir une base solide sur la compilation
- Reconnaissance de forme
- Etre ingénieu

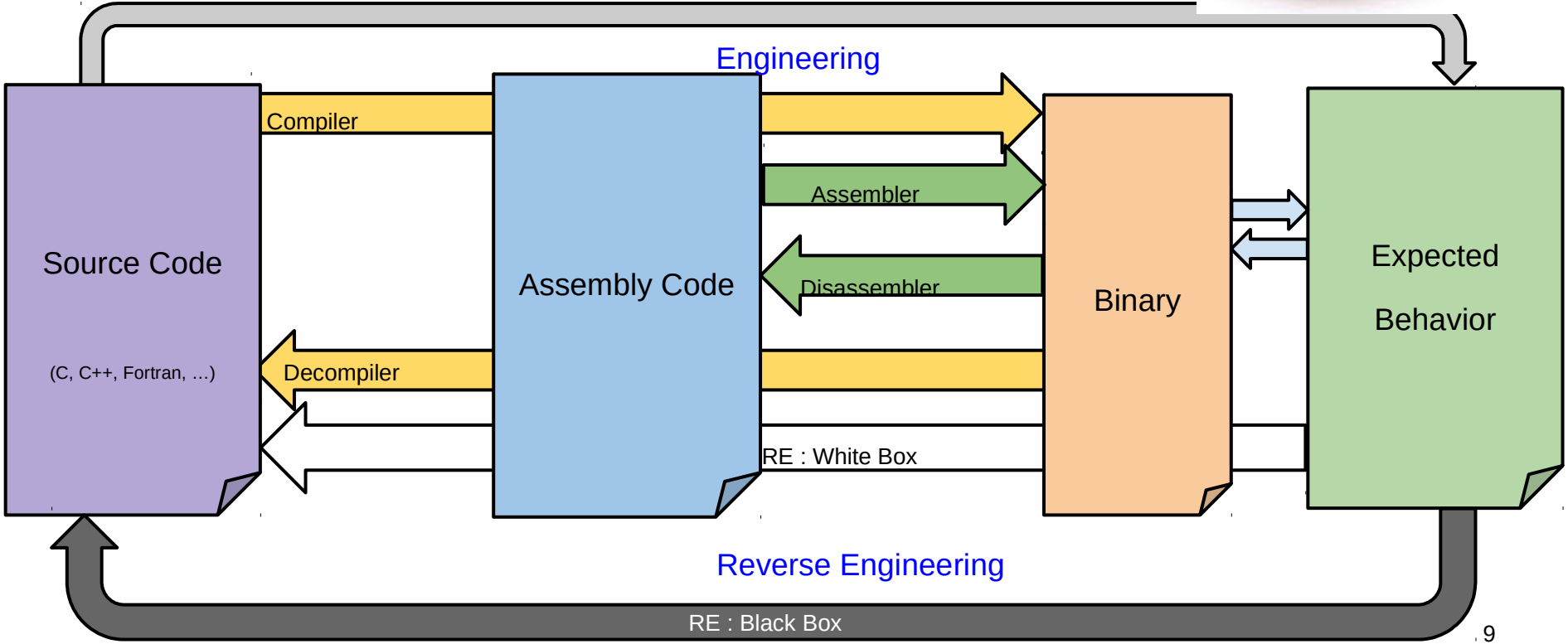
# Introduction - Compétences

- Processeur? Ca fait quoi un processeur?
- Plein de choses fabuleuses. Voici quelques optimisations :
  - Prédiction de branchement
  - Déroulement de boucle
  - Alignement
  - Inlining (chrome, noyau windows, ...)

# Introduction - Compétences

- Et la protection logicielle?
- 3 grands types :
  - Protection matérielle (dongle)
  - Numéro de série / licence
  - Activation en ligne

# Concepts - Présentation



# Concepts - Introduction

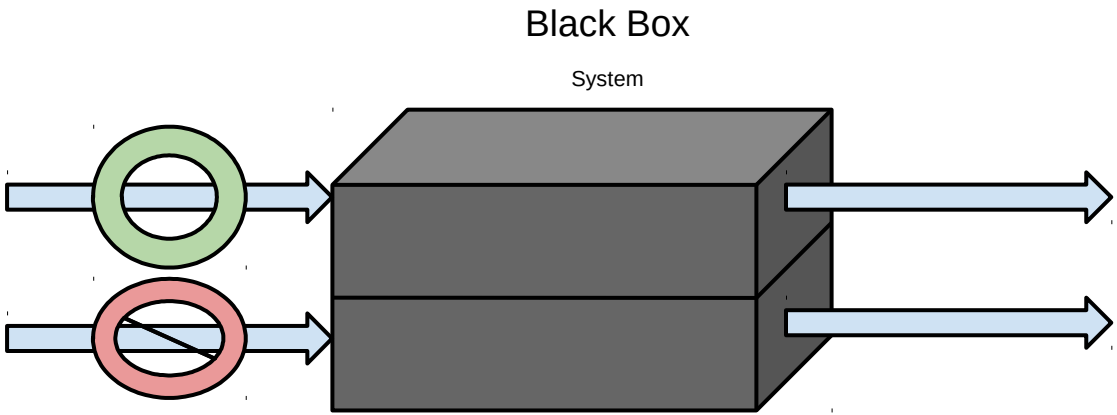
- Contexte d’analyse :
  - Boîte Blanche et Boîte Noire
  - Statique et Dynamique
  - Actif et Passif

# Concepts - Architecture

- Boite Noire :
  - Dynamique - Accès au produit pendant son fonctionnement
    - Passif : Pas d’interaction avec les entrées / sorties
    - Actif : Interaction

# Concepts - Architecture

- Boite Noire :

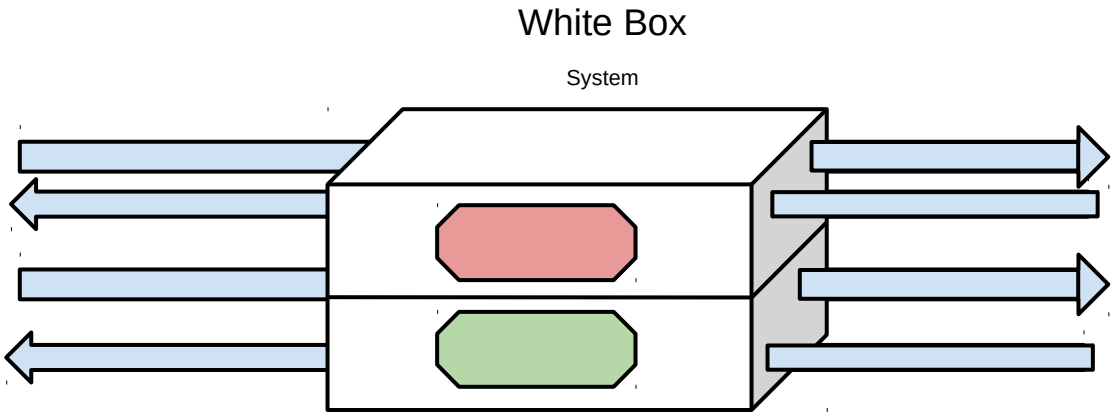


# Concepts - Architecture

- Boite Blanche :
  - Dynamique - Accès au produit pendant son fonctionnement
  - Statique - Analyse indépendante de toute exécution

# Concepts - Architecture

- Boite Blanche :



# Concepts - Abstractions

Niveaux d’abstractions :

- Application
  - Concepts de l’application et de son domaine
- Fonction et logique

# Concepts - Abstractions

- Structure et données
  - Architecture du système
- Implémentation
  - Table des symboles
  - Chaînes de caractères

# Outils

- Comment faire du reverse?
  - OS spécialisé : Kali linux
  - Analyse réseaux
    - netstat
    - nmap
    - et bien d’autres

# Outils

- Analyse de binaire
  - strings - liste des strings
  - nm - liste des symboles
  - ltrace - liste des appels de librairies
  - strace - liste des appels systèmes
- Décompilateur
  - boomerang
  - Hex-Rays IDA

# Outils

- Désassembleur
  - objdump
  - Ollydbg
- Editeur hexadécimal
  - Bless
  - Emacs avec hexl-mode
  - hte (analyseur d’executable, …)

Et bien d’autres!!!

# Techniques

- Boîte Noire :
  - Active
    - Fuzzing
      - Entrées générées aléatoirement,
      - Adapter les entrées par rapport aux sorties
  - Passive
    - Analyse des trames

# Techniques

- Boîte Blanche :
  - Statique
    - Interprétation abstraite
      - Approximation de l’ensemble des valeurs des variables
  - Dynamique
    - Fuzzing intelligent : exécution symbolique dynamique
    - Tests de couverture
    - Tests unitaires

# Démo

- Boîte Blanche :
  - Bypass d’un binaire
- Boîte Noire :
  - Fuzzing sur un serveur

# Conclusion

Reverse Engineering :

- Nombreux outils et techniques
- Utilisé dans de nombreux contextes
- Cible essentiellement la sécurité des systèmes
- Nécéssite des bases solides

# Questions?