IDArling

La première plateforme de rencontre entre reversers

Plugin de reverse collaboratif pour IDA Pro et Hex-Rays

https://github.com/IDArlingTeam/IDArling

SSTIC 2019

QUI SOMMES-NOUS ?

- Alexandre Adamski
  - Ingénieur R&D @ Quarkslab
- Joffrey Guilbon
  - Ingénieur R&D @ Quarkslab

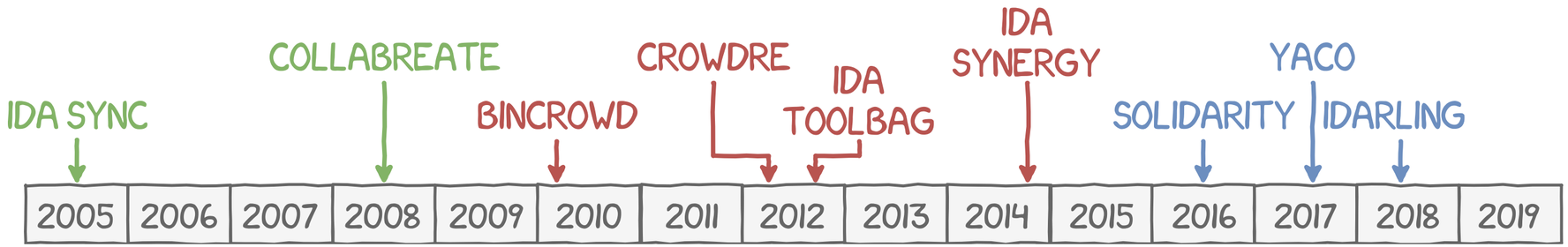
PLAN DE LA PRÉSENTATION

- Démonstration
- Historique et motivations
- Améliorations et difficultés
- Alternatives et conclusion

DÉMONSTRATION

HISTORIQUE ET MOTIVATIONS

HISTORIQUE



Deux paradigmes : temps réel / gestion de versions

POUR FAIRE DU TEMPS RÉEL

collabREate

- Développé par Chris Eagles
- Stable et support IDA 7.0
- Écrit en C++ (compilation)
- Manque d'évolutions

Sol[IDA]rity

- Écrit en Python
- Nouvelles fonctionnalités
- Indisponible au public
- Dépendances externes

IDÉE : POURQUOI NE PAS REFAIRE SOL[IDA]RITY EN OPEN-SOURCE ?

Liste des ingrédients :

- Slides et vidéo de la RECON 2016
- Site officiel et ses GIFs : https://solidarity.re
- De la motivation et beaucoup de soirées libres
- Des bêta-testeurs (merci panda/karion/nezetic/p0ly)

AMÉLIORATIONS ET DIFFICULTÉS

AMÉLIORATION : NETWORKING

- Sol[IDA]rity utilisait le framework Twisted
  - Introduit une autre *event loop*
- Il existe qt-reactor pour s'intégrer dans Qt5
  - Une dépendance de plus, buggé
- QtNetwork, la framework de *networking* natif de Qt
  - Pas inclus de base dans IDA
- Écriture de zéro d'un système de paquets intégré dans l'*event loop* de Qt en utilisant des `QSocketNotifier`

AMÉLIORATION : INSTALLATION

- Ne pas avoir de dépendance externe permet une installation
  - par drag-and-drop dans le dossier `plugins`
  - en entrant un one-liner dans le console Python d'IDA
    - adapté du `install_from_ida` d'ipyida
- Pour installer le serveur, besoin uniquement de PyQt5 :
  - `pip install PyQt5` pour Python 3

AMÉLIORATION : SERVEUR

- Intégration du serveur directement dedans IDA
  - Serveur dédié : script Python externe
  - Serveur intégré : démarré depuis l'interface d'IDA
- **Problème :** le serveur intégré utilise un port aléatoire
- **Solution :** découverte automatique de serveurs sur le LAN
  - Utilise des paquets UDP envoyés à intervalle régulier

DIFFICULTÉ : CHARGER UNE IDB

- Pas de fonction dans IDA pour changer d'IDB
- Solution trouvée par *trial and error* :
  - `dll = self._plugin.core.get_ida_dll(app_name)`
  - `dll.term_database()`
  - `dll.init_database(argc, argv, pv)`
  - `shutil.copyfile(file_path, tmp_path)`
  - `ida_kernwin.restore_database_snapshot(s, None, None)`
  - `os.remove(tmp_path)`

DIFFICULTÉ : COLORER UNE FONCTION

- Changer la couleur d'une fonction avec `func.color = xxx`
- Sauf que la couleur est visible dans la *Disassembly View*…
- Solution trouvée par *trial and error* :
  - récupérer le widget Qt sous-jacent avec SIP
  - ajouter un proxy sur l'`ItemModel`
  - ajouter un proxy sur l'`ItemDelegate`
  - installer un `EventFilter` pour les *tooltip*
- `EventFilter` aussi utilisé pour le menu contextuel

DIFFICULTÉ : BINDINGS PYTHON CASSÉS

- Les bindings Python sont soit cassés, soit non existants
  - ajouter un `ida_idp.IDP_Hooks` et aller dans *Options > Compiler* provoque un crash avec IDA 7.2
  - le binding de `ida_typeinf.set_numbered_type` est incorrect, obligé d'utiliser `ctypes`
  - pas possible de changer la valeur de retour de `ev_get_bg_color` sans `ctypes`
  - les bindings d'Hex-Rays ne sont pas en reste
- Un exemple poussé à l'extrême : MCExplorer

ALTERNATIVES ET CONCLUSIONS

QUE FAIRE EN CAS DE COLLISION ?

- Modèle dans lequel l'utilisateur est toujours connecté
- Désynchronisations mineures sont ignorées
- Désynchronisations majeures ne sont pas gérées
  - implique de connaître l'inverse de chaque action
  - ou de pouvoir restorer l'IDB dans un état antérieur

OBJECTIFS FINAUX

- Synchronisation temps réel des IDB
- Synchronisation temps réel de Hex-Rays
- Rejeu des évenements manqués par un utilisateur
- Affichage en temps réel des utilisateurs sous forme de curseurs (navbar, désassemblé, fonction)
- Gestion des projets et des utilisateurs

I ~~HAVE~~ HAD A DREAM

- Synchronisation entre IDA / Binary Ninja / etc.
- Implémenter un vrai UNDO dans IDA
  - Idée : hooker les modifications aux *netnodes*

PAR RAPPORT À YACO

YaCo	IDArling
• Synchronisation à la demande • Supporte le travail déconnecté • Gestion des conflits	• Synchronisation temps réel • Besoin d'être tout le temps connecté • Interactions entre les utilisateurs

IDArling  YaCo

CONCLUSION

- Produit minimum viable qui fonctionne
- Encore beaucoup d'améliorations à faire
- Un projet qui nous a énormément appris
- Recherche de contributeurs pour nous aider