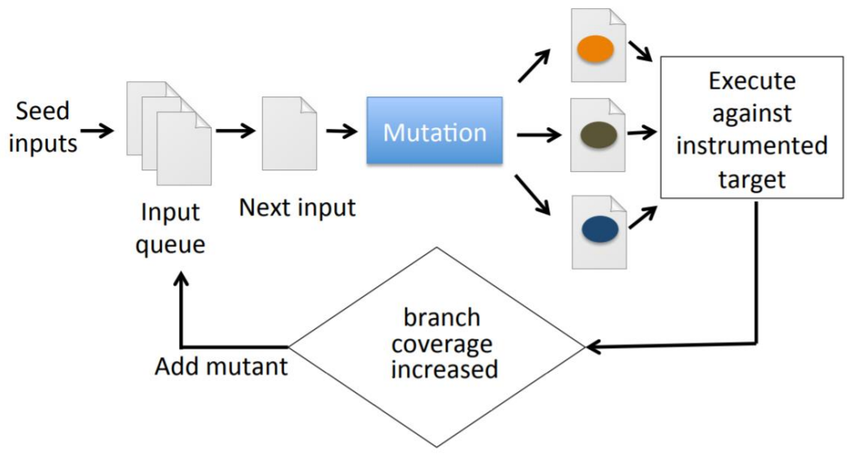
Outil de fuzzing en black-box pour cibles embarquées

Auteurs : LAGACHE Zoé – LECOMTE Benjamin – VAREILLE Simon

CONTEXTE ET OBJECTIF

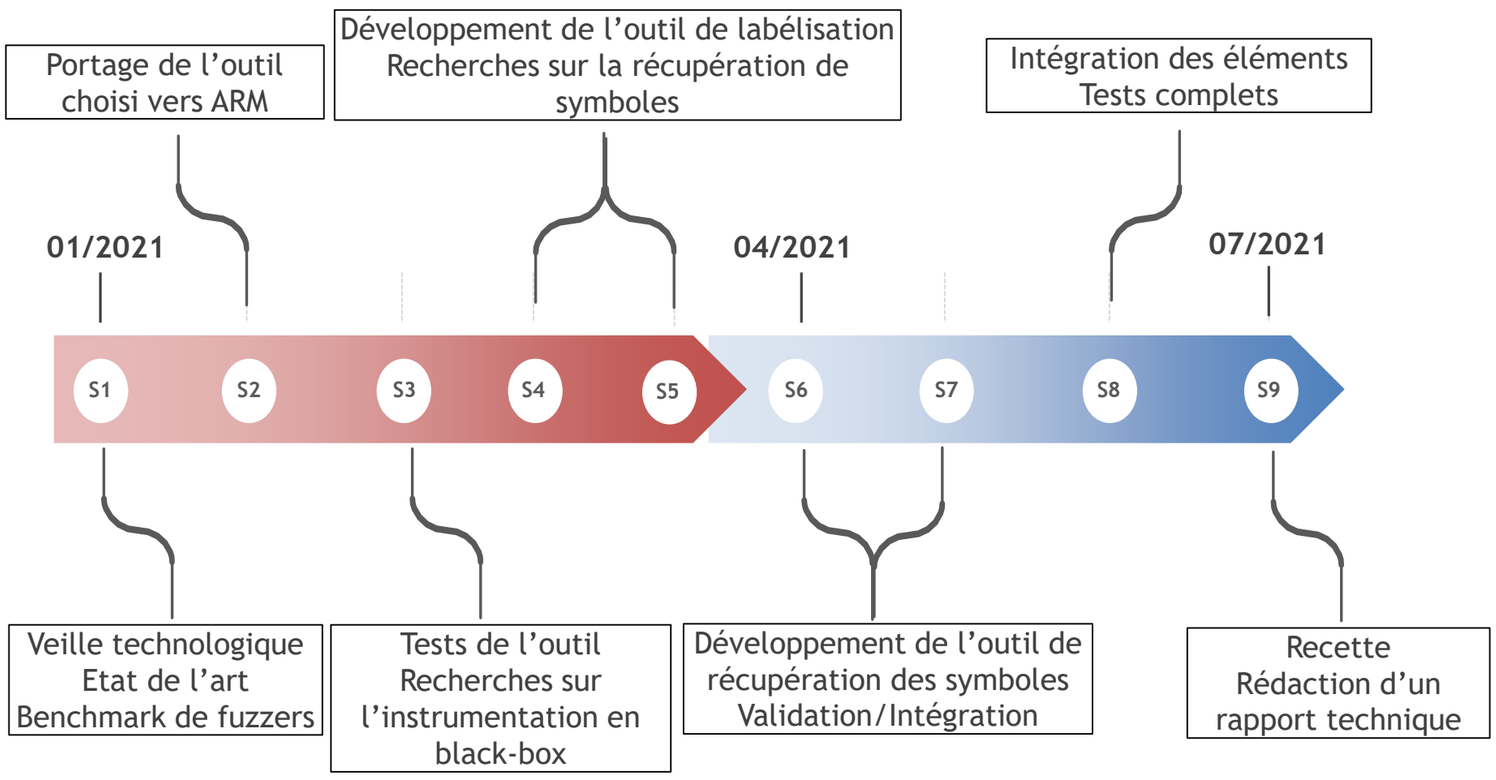
Le programme **Pulse** de l'**IRT Nanoelec** vise à développer la cybersécurité, notamment dans le domaine de l'IoT. La cybercriminalité se développant rapidement, il devient important pour les partenaires industriels d'intégrer la sécurité dans leurs cycles de développement.



Le **fuzzing** est une technique de test qui génère automatiquement une grande quantité de données d'entrée pour stimuler un programme dans le but de **détecter** des comportements imprévus et des **vulnérabilités**.
L'objectif de ce projet est de créer un outil pour appliquer les méthodes de fuzzing en **black-box** (lorsque le code source de l'application testée n'est pas disponible) sur des programmes embarqués.

MÉTHODES ET DÉVELOPPEMENTS

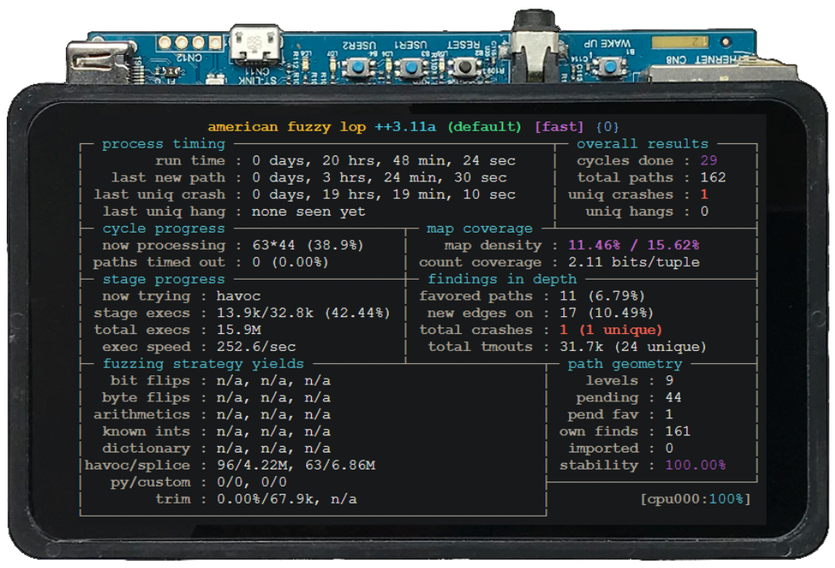
Le projet a suivi la méthode agile SCRUM.
Cette méthode a permis d'avoir un retour rapide sur l'orientation du projet et sur les différentes tâches effectuées.
En particulier, le travail a été divisé en **sprints** de deux semaines.



L'outil de fuzzing en black-box a été développé en **C** et en **Python**.
Il se base également sur des outils de reverse comme **Ghidra** et **radare2**.

RÉSULTATS ET CONCLUSION

Le projet a démontré qu'il est possible de faire du fuzzing en **black-box** sur cibles embarquées. De plus, il est possible de conserver de bonnes performances en passant par une **instrumentation statique** plutôt que dynamique.
Cet outil peut désormais être utilisé pour automatiser la recherche de **vulnérabilités** dans des applications embarquées.



L'outil développé permet notamment :
- ✓ La récupération des symboles de binaires ELF strippés
- ✓ Le désassemblage et la labélisation de binaires
- ✓ L'instrumentation de l'assembleur récupéré
- ✓ Le fuzzing de l'application par le fuzzer choisi

MOTS-CLÉS : Fuzzing, black-box, ARM, réécriture de binaires, instrumentation statique