Etude et développement d'un Framework de Fuzzing réseau

Auteurs : Kevin Guiguet – Loup Doinel – Gabin Chognot

CONTEXTE ET OBJECTIF

Safran Electronics & Defense est l'une des dix entités groupe international Safran, plus ancien groupe industriel aéronautique au monde, opérant également dans les domaines de l'espace et de la défense.

Safran E&D a besoin de développer des outils de test afin de s'assurer de la sureté et du bon fonctionnement de ses équipements et de ses systèmes en cas d'attaque potentielle.

Développement d'un outillage qui, à partir de données nominales, automatisera la génération de trames pseudo-aléatoirement (Fuzzing), les enverra à l'équipement et assurera le monitoring de la cible afin d'identifier des comportements non définis.

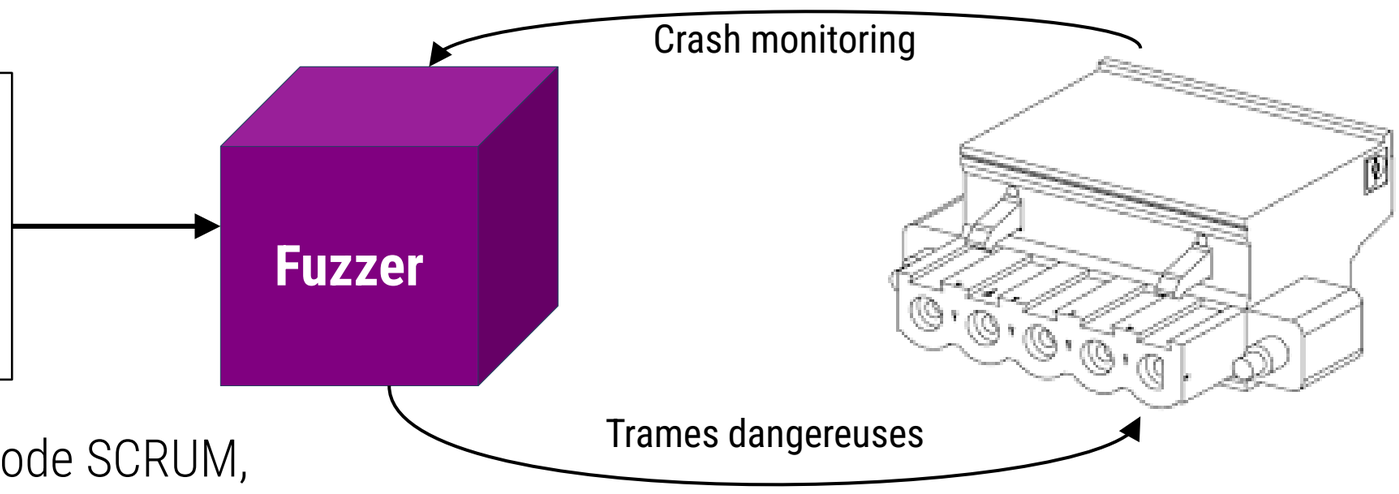
MÉTHODES ET DÉVELOPPEMENTS

L'état de l'art (*SOTA*), puis la rédaction de documents de *specs* et de *benchmark* ont permis, d'une part le choix d'un outil déjà existant comme base de travail, Boofuzz, et l'écriture d'un document de *backlog*, contenant l'ensemble des fonctionnalités à y ajouter.

Input : Spécification de protocole

Type	Static	String	Byte	Static	Byte
Name	opcode	filename	null	mode	null

TFTP

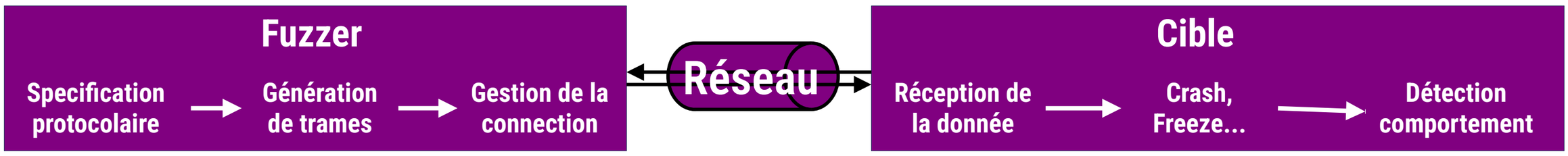


Ce projet a été conduit au travers d'une méthode SCRUM, découpé en 2 *releases* de 4 puis 3 *sprints* respectivement, pour lesquels Safran attribuait à l'équipe projet les fonctionnalités à développer pour le *sprint* (deux semaines).

Développement Python, versioning Git, containeurisation Docker, documentation Sphinx.

RÉSULTATS ET CONCLUSION

Le fuzzer, à partir, d'une spécification de protocole, génère des trames dont les champs contiennent des valeurs issues de seclists, d'autres mutées, et d'autres générées aléatoirement. Il implémente l'ensemble des fonctionnalités nécessaires au support de protocoles grands publics (HTTP, FTP, TFTP …) et aéronautiques.
Le fuzzer envoie les trames en supportant diverses piles protocolaires (TCP, UDP, WebSocket…), et monitor la cible (délai et contenu de la réponse, utilisation des ressources systèmes, PID…).



MOTS-CLÉS : Fuzzing, Tests de robustesse, Protocoles aéronautiques, Python