الجمهورية الشعبية الديمقراطية الجزائرية
République Algérienne Démocratique et Populaire
وزارة التعليم العالي و البحث العلمي
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
المدرسة العليا للإعلام الآلي ٠08 ماي 1945٠ بسيدي بلعباس
École Supérieure en Informatique
-08 Mai 1945- Sidi Bel Abbès

Mémoire de Fin d'étude

En Vue de l'obtention du diplôme de **Master**
Filière : **Informatique**
Spécialité : **Ingénierie des Systèmes Informatiques (ISI)**

Thème

Detecting and Exploiting Binary Vulnerabilities using Symbolic Execution Engines

---

Présenté par :

- Tasfaout Abderrahim

Soutenu le : **04/10/2021**

Devant le jury composé de :

Mr BENDAOUAD Fayssal	Docteur	Président
Mr Alaa Eddine Belfedhal	Docteur	Encadreur
Mme ANANI Djihed	Docteur	Examinatrice

Année Universitaire : 2020/2021

Abstract

Bugs in software have always been a problem to developers and especially security-critical bugs that can make the software vulnerable. In the case of exploitable vulnerabilities, attackers can use them to leak data, attack systems with denial of system attacks, or in the worst-case gain full access to the system. Many of these vulnerabilities are in binaries. Therefore, it is hard for developers and testers to detect them and fix them.

Many previous articles were about finding a way to automate the process of detecting binary vulnerabilities and, we will focus on approaches that used symbolic execution and compare between them. Then, we will see the results provided by the article plus, showing what can be performed to improve the binary vulnerability detection process.

Résumé.

Les bugs dans les logiciels ont toujours été un problème pour les développeurs, en particulier les bugs critiques pour la sécurité qui peuvent rendre le logiciel vulnérable. Dans le cas de vulnérabilités exploitables, les attaquants peuvent les utiliser pour faire fuir des données, attaquer des systèmes par Les attaques DOS ou, dans le pire des cas, obtenir un accès complet au système. Un grand nombre de ces vulnérabilités se trouvent dans des binaires. Il est donc difficile pour les développeurs et les testeurs de les détecter et de les corriger. De nombreux articles précédents visaient à trouver un moyen d'automatiser le processus de détection des vulnérabilités binaires. Nous allons nous concentrer sur les approches qui utilisent l'exécution symbolique et les comparer. Ensuite, nous conclurons quelles approches ont été les meilleures dans le processus de détection et nous montrerons ce qui peut être fait pour améliorer le processus de détection des vulnérabilités binaires.]