IX ENDICT
Encontro de iniciação científica
Campus Toledo
2023

Gerador de Chaves Presentes em Instruções de Máquina por Teoria de Satisfabilidade do Módulo

José Luis Bastos Donin[1]
donin@alunos.utfpr.edu.com
Orientador:
Daniel Cavalcanti Jeronymo

[1]*Universidade Tecnológica Federal do Paraná – Campus Toledo Programa de IC*

RESUMO: O objetivo deste projeto é aplicar as Teorias de Módulos de Satisfatibilidade (SMT - *Satisfiability Modulo Theories*) à geração automática de chaves para acesso a sistemas através da análise de arquivos binários. Arquivos binários contém instruções de máquina que são carregadas na memória para execução por um processador (CPU – *Central Processing Unit*) e geralmente não possuem conversão reversa simples para o código de programação usado para gerar as instruções. O desenvolvimento de um sistema que automatize o processo de engenharia reversa através do *Binary Analysis and Reverse Engineering Framework* (BARF) objetiva traduzir as instruções de máquina da plataforma Intel x86 para uma linguagem intermediária independente de plataforma chamada *Reverse Engineering Intermediate Language* (REIL). O Z3-solver é um provador de teoremas criado pela Microsoft com algoritmos especializados para verificação e execução simbólica, facilitando a aplicação de SMT em problemas mais gerais. O algoritmo proposto realiza uma análise no arquivo executável PE32 (exe) no Windows ou arquivo executável ELF no Linux fornecido pelo usuário sem o código-fonte, o usuário deve especificar os pontos de entrada e saída dos endereços de memória a serem analisados pela plataforma, as instruções entre esses pontos são traduzidas das instruções de máquina para a linguagem intermediária através do BARF. As instruções traduzidas são analisadas pelo Z3-solver a fim de determinar a satisfatibilidade das restrições através da execução simbólica. Na primeira etapa de teste, foi criado um programa de validação em C no qual o usuário insere um número de cinco dígitos, os dígitos são iterados através de um laço de repetição e validados por adição e restos, se o resto da soma dos dígitos for zero, a chave é válida. Um programa python também foi criado usando apenas o Z3-solver onde as restrições foram adicionadas manualmente para testar sua utilidade no processo de engenharia reversa. Ao adicionar manualmente as restrições foi possível gerar uma chave válida para o programa C, mesmo sem o código-fonte original. De acordo com os resultados preliminares, é possível gerar uma chave de validação adicionando manualmente as restrições ao Z3-solver, sendo capaz de gerar uma chave de validação desde que as restrições sejam satisfatórias. Trabalhos futuros incluem aplicar os resultados ao adaptar a abordagem usando BARF e REIL para automatizar o processo de geração de chaves pela análise das instruções traduzidas para REIL. Os principais desafios enfrentados atualmente são a falta de compatibilidade com as versões atuais do Python, pois é necessário adaptar as tecnologias disponíveis para versões atualizadas.

Palavras-chave: análise binária, chaves de acesso, chaves criptográficas, teorias de satisfabilidade do módulo, engenharia reversa.

ABSTRACT: The objective of this project is to apply Satisfiability Modulo Theories (SMT) to the automatic generation of keys for accessing systems through the analysis of binary files. Binary files contain machine instructions that are loaded into memory for execution by a processor (CPU – Central Processing Unit) and generally do not have simple back conversion to the programming code used to generate the instructions. The development of a system that automates the reverse engineering process through the Binary Analysis and Reverse Engineering Framework (BARF) aims to translate machine instructions from the Intel x86 platform into a platform-independent intermediate language called Reverse Engineering Intermediate Language (REIL). Z3-solver is a theorem prover created by Microsoft with specialized algorithms for symbolic verification and execution, facilitating the application of SMT to more general problems. The proposed algorithm performs an analysis on the PE32 executable file (exe) on Windows or ELF executable file on Linux provided by the user without the source code, the user must specify the entry and exit points of the memory addresses to be analyzed by the platform, the instructions between these points are translated from the machine instructions to the intermediate language through BARF. The translated instructions are analyzed by Z3-solver in order to determine the satisfiability of the constraints through symbolic execution. In the first test stage, a validation program was created in C in which the user enters a five-digit number, the digits are iterated through a repetition loop and validated by addition and remainders, if the remainder of the sum of the digits is zero, the key is valid. A python program was also created using just Z3-solver where constraints were added manually to test its usefulness in the reverse engineering process. By manually adding the restrictions it was possible to generate a valid key for the C program, even without the original source code. According to the preliminary results, it is possible to generate a validation key by manually adding the constraints to Z3-solver, being able to generate a validation key as long as the constraints are satisfactory. Future work includes applying the results by adapting the approach using BARF and REIL to automate the key generation process by analyzing instructions translated into REIL. The main challenges currently faced are the lack of compatibility with current versions of Python, as it is necessary to adapt available technologies to updated versions.

Keywords: binary analysis, access keys, cryptographic keys, satisfiability modulo theories, reverse engineering.