Static Detection Techniques for Identifying Packed Executable Binaries

Executable packing consists of applying transformations to binary executables, including compression, encryption, and code virtualization. Even though this method of altering a binary file can be useful to protect software from piracy and for license management, it is also widely used on malware in order to evade detection systems. Therefore, packing still remains challenging for malware analysis today. Over the two last decades, many studies from the scientific literature addressed various techniques for detecting packing with varying degrees of accuracy, leveraging entropy, risk score based heuristics or also machine learning models. While static techniques evolved progressively towards machine learning based techniques with tailored features, most of these studies focused on the Portable Executable format (i.e. used in Windows) and not on other formats like Executable and Linkable Format (ELF - i.e. used in Linux).

In 2023, an experimental toolkit[1] called the Packing Box was released and presented by D'Hondt[2] for gathering a comprehensive set of tools allowing to test many state-of-the-art static detectors and training models while tuning each step of the machine learning pipeline. This toolkit was designed to support multiple executable formats in the hope of building better and more generalized models. In particular, it supports ELF binaries and already integrates a few packers that can be used to build datasets thanks to an automation allowing for mass-packing.

Project Objectives:

- Take advantage of the Packing Box and study the packing of ELF/APK binaries and its static detection, parsing and complementing the current literature[3][4].
- Recent binary packers shall be integrated to the Packing Box to extend its panel of packers.
- Detection techniques, including machine learning models fine-tuned through feature engineering, shall be compared according to their performance and accuracy.

Profile/Required Skills:

- Reverse Engineering and Binary Analysis
- Python Programming, Machine learning, Scikit-learn Framework

Practicalities

Promotor:	Prof. Bart Preneel		
Daily supervisor:	Enrique Argones Rúa	enrique.argonesrua@esat.kuleuven.be	B01.13
	Alexandre D'Hondt	alexandre.dhondt@uclouvain.be	
	Irfan Bulut	irfan.bulut@esat.kuleuven.be	B01.28
	Charles-Henry Bertrand Van Ouytsel	charles-henry.bertrand@uclouvain.be	
	Ramin Sadre	ramin.sadre@uclouvain.be	

Nature of the work:	20% Literature, 20% Theoretical work, 30% Software Development, 30% Experiments
Number of students:	1 or 2

[1] https://github.com/packing-box/docker-packing-box
[2] https://dl.acm.org/doi/10.1007/978-3-031-61231-2_17
[3] https://link.springer.com/article/10.1007/s10115-011-0393-5
[4] https://ieeexplore.ieee.org/document/11012083

 © 2026 – 2027 COSIC