Detecting the Misuse of Cryptographic Methods with Machine Learning

Gustavo Eloi de P. Rodrigues[1], **Ricardo Dahab**[1], **Alexandre M. Braga**[1]

[1]Instituto de Computação – Universidade Estadual de Campinas (Unicamp)

g230218@dac.unicamp.br, rdahab@ic.unicamp.br, alexmbraga2007@gmail.com

Abstract. *Misuse of cryptographic libraries leads to vulnerabilities that, if exploited, can cause problems related to application data confidentiality, integrity, and availability, causing major harm to software developers. Therefore, this research project proposes the use of machine learning for detecting cryptographic library misuse in combination with source code representations that preserve syntactic, semantic, and data-flow structures. Additionally, this project aims to indicate in the source codes where the detected misuse occurs.*

Resumo. *Mau-usos de bibliotecas criptográficas levam a vulnerabilidades que, se exploradas, podem causar problemas relacionados à confidencialidade, integridade e disponibilidade de dados de aplicações, causando grandes prejuízos aos desenvolvedores de software. Sendo assim, este projeto de pesquisa propõe o uso de aprendizado de máquina para a detecção de maus-usos de bibliotecas criptográficas em combinação com representações de códigos-fonte que preservem estruturas sintáticas, semânticas e de fluxo de dados. Adicionalmente, este projeto visa indicar nos códigos-fonte o local em que ocorre os maus-usos detectados.*

1. Introduction

With the increasing use of technologies and applications that require assurance of requirements such as data integrity, confidentiality, authenticity and availability, encryption has been increasingly used for this purpose. However, most developers responsible for building these applications have limited knowledge of cryptography. In addition, the cryptographic libraries used are not easy to understand and have limited documentation, which results in the difficulty of their use and often leads to their misuse [Lazar et al. 2014], which causes cryptographic misuses.

As a result, many companies rely on tools to aid in the development of cryptographic applications. However, only 30% of these misuses are detected on average by these tools and the combination of two or more of them detects a maximum of 50%, leading to the introduction of software vulnerabilities in these applications [Braga et al. 2017]. Therefore, solutions that effectively support application developers to incorporate encryption simply and effectively into their applications are urgently needed [Nadi et al. 2016].

2. Goals

In order to improve the detection of such misuse, we propose an approach that uses machine learning to detect misuse rather than the matching of predefined patterns and rules employed by the available tools. Thus, the main expected goals of this project are:

- Develop a machine learning model capable of identifying misuse of cryptographic libraries through the use of source code representation structures.
- Classify detected misuse according to defined categories.
- Indicate in the source code analyzed where a detected misuse occurs.

3. Methodology

Our approach will be divided into four phases:

1. **Data Collect**: Collect source codes (data) from other works such as [Braga and Dahab 2016] and others.
2. **Feature Engineering**: Transform source code into Abstract Syntactic Trees and Program Dependency Graphs and extract features using the Bag of Graphs [Silva et al. 2014] method.
3. **Train and Validation of Classifiers**: Train and Test of various classifiers using specific metrics.
4. **Test and choice of Classifiers**: Test of trained classifiers and choice of best ones.

4. First Results, Conclusion and Next Steps

At the time of this poster, we have assembled a dataset of approximately 19,000 cryptographic misuse source codes that are yet to be categorized according to [Braga and Dahab 2016]. Collect data is difficult due the lack of datasources. However, this work is only at the beginning of its development, but with great prospects The next steps will be to implement the Bag of Graphs [Silva et al. 2014] methods for vectoring.

5. Acknowledgements

We thank CAPES (Coordination for the Improvement of Higher Education Personnel) and LASCA (Laboratory of Security and Cryptography) for all support.

References

Braga, A. and Dahab, R. (2016). Mining cryptography misuse in online forums. In *2016 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C)*, pages 143–150. IEEE.

Braga, A., Dahab, R., Antunes, N., Laranjeiro, N., and Vieira, M. (2017). Practical evaluation of static analysis tools for cryptography: Benchmarking method and case study. In *2017 IEEE 28th International Symposium on Software Reliability Engineering (ISSRE)*, pages 170–181. IEEE.

Lazar, D., Chen, H., Wang, X., and Zeldovich, N. (2014). Why does cryptographic software fail?: a case study and open problems. In *Proceedings of 5th Asia-Pacific Workshop on Systems*, page 7. ACM.

Nadi, S., Krüger, S., Mezini, M., and Bodden, E. (2016). Jumping through hoops: Why do java developers struggle with cryptography apis? In *Proceedings of the 38th International Conference on Software Engineering*, pages 935–946. ACM.

Silva, F. B. et al. (2014). Bag of graphs = definition, implementation, and validation in classification tasks. *Repositório Unicamp. Universidade Estadual de Campinas, Instituto de Computação, Campinas, SP. Available in: http://www.repositorio.unicamp.br/handle/REPOSIP/275527. Access in: 25 aug. 2018.*