Short Paper: Toward Bridging The Gap Between Black-box Deobfuscation and Program Synthesis*

Grégoire Menguy, Vidal Attias, Nicolas Bellec,
Frédéric Recoules, and Sébastien Bardin

Université Paris Saclay, CEA, List, Palaiseau, France
first.last@cea.fr

Abstract. Xyntia is a state-of-the-art of black-box deobfuscator, which aims to simplify highly obfuscated code snippets, i.e. code that is artificially complexified to prevent reverse and analysis. Black-box deobfuscation relies on observed input-output behaviors to synthesize a simpler version of the obfuscated code. While there is a clear link with formal program synthesis, the two communities do not really interact, possibly due to mutual ignorance and a lack of common benchmarks and usage scenarios. The goal of this talk is twofold: (i) to start building bridges between the synthesis and the deobfuscation communities; and (ii) to give the first synthetic and comprehensive presentation of Xyntia, along with the first presentation of its sampler. Especially, we provide the necessary material to run Xyntia on synthesis benchmarks, and to easily apply synthesis tools on real deobfuscation cases.

Keywords: Deobfuscation · Program synthesis · Program analysis.

1 Introduction

Syntax-guided program synthesis (SyGus) [1] aims to generate code from a logical and a syntactical specification. The former specifies the expected code behavior, and the latter narrows the search space to a specific language. SyGus has been extensively studied in the Programming Language and Formal Methods communities [12,1,8,11], with many applications from coding assistant [8] to invariant synthesis [15].

Concurrently, black-box deobfuscation recently emerged in the system security community [6,5,13,3]. The goal here is to revert obfuscations [10,9] i.e., code transformations used to impede reverse engineering. Deobfuscation is especially useful to analyze malware that are often highly obfuscated [16,17]. Black-box deobfuscators consider the obfuscated code snippet as an oracle and rely on observed input-output behaviors to recover a simpler version of it. Being black-box, they bypass standard obfuscations such as MBA [20] or covert channels [18].

* This work has benefited from grants managed by ANR under France 2030 with references ANR-22- PECY-0007 and by BPI under Plan France 2030 with reference DOS0233319/00.

Interestingly, formal program synthesis and black-box deobfuscation are two very similar problems. Still, no deep link has been made between the two communities. Each relies on different synthesis methods: SyGus synthesizers often perform enumerative search [4,11,8], while black-box deobfuscators focus on stochastic search [5,13]. No conceptual comparison have been performed between these different methods. None of them has been evaluated against the others community use cases. We believe that it comes from a missing analysis of each community's specificities, and missing tools to apply synthesizers and black-box deobfuscators to the different usecases and datasets.

Goal of the talk. We present our efforts to build bridges between the synthesis and deobfuscation communities. This includes the first evaluation of black-box deobfuscation and SyGus synthesizers on deobfuscation and SyGus benchmarks. To better compare the different approaches we also provide the first synthetic and comprehensive overview of the Xyntia black-box deobfuscator [13,3], including its new sampler. Xyntia is one of the seminal black-box deobfuscators (along with Syntia, no more maintained). It delievers state of the art performance and even outperforms some dedicated white-box deobfuscators on their specific scenarios [13,3]. More precisely, the talk will describe:

- The *Xyntia Sampler*, a flexible sampling framework to help the synthesis community apply their synthesizers to black-box deobfuscation. This sampler can be plugged into any synthesizer accepting the standard sygus2 format [14]. To demonstrate its simplicity of use, we combined it with Cvc4/5 [4] and automatically analyzed a binary obfuscated with VMprotect [19] and the Xtunnel malware [7];
- The *Xyntia Synthesizer*, its options and features, including its grammars, and objective functions. We also highlight recent improvements, including the introduction of **(i)** inference rules to handle usually hard-to-synthesize expressions; **(ii)** a more flexible grammar definition; and **(iii)** a CEGIS loop [1]. These improvements aims to improve Xyntia and ease its application on deobfuscation and beyond. Experiments show their complementarity;
- The evaluation of Xyntia against the Cvc4/5 synthesizer over both deobfuscation datasets and a benchmark from the SyGus competition [2]. Experiments show (without surprise) that each tool outperforms the other on its own domain of application. Yet, they also show that Cvc4/5 is prone to overfitting, relying too much on conditional operators, which may explain its good results over SyGus problems. Moreover, results show that the deobfuscation use-case is a great opportunity to bypass the usual SyGus limitations and devise new synthesis methods. Especially, the CEGIS and our new preprocess help Xyntia on complex cases, respectively taking advantage of the available smt ground-truth and executable oracle.

Tool and access. We provide a toolset for synthesis-based deobfuscation, including: **(i)** a state-of-the-art programming-by-example engine for the Bitvector theory and; **(ii)** a flexible sampler to apply synthesizers over real obfuscated binaries. All is publicly available at https://github.com/binsec/xyntia.

References

1. Rajeev Alur, Rastislav Bodík, Garvit Juniwal, Milo M. K. Martin, Mukund Raghothaman, Sanjit A. Seshia, Rishabh Singh, Armando Solar-Lezama, Emina Torlak, and Abhishek Udupa. Syntax-guided synthesis. In *Formal Methods in Computer-Aided Design, FMCAD 2013*. IEEE, 2013.
2. Rajeev Alur, Dana Fisman, Saswat Padhi, Rishabh Singh, and Abhishek Udupa. Sygus-comp 2018: Results and analysis. 2019.
3. Vidal Attias, Nicolas Bellec, Grégoire Menguy, Sébastien Bardin, and Jean-Yves Marion. Augmenting search-based program synthesis with local inference rules to improve black-box deobfuscation. In *Proceedings of the 2025 ACM SIGSAC Conference on Computer and Communications Security*, CCS '25, 2025.
4. Clark Barrett, Christopher L. Conway, Morgan Deters, Liana Hadarean, Dejan Jovanovi'c, Tim King, Andrew Reynolds, and Cesare Tinelli. CVC4. In Ganesh Gopalakrishnan and Shaz Qadeer, editors, *Proceedings of the 23rd International Conference on Computer Aided Verification (CAV '11)*, 2011.
5. Tim Blazytko, Moritz Contag, Cornelius Aschermann, and Thorsten Holz. Syntia: Synthesizing the semantics of obfuscated code. In *Usenix Security*, 2017.
6. Tim Blazytko, Moritz Contag, Cornelius Aschermann, and Thorsten Holz. Syntia: Breaking state-of-the-art binary code obfuscation via program synthesis. *Black Hat Asia*, 2018.
7. J Calvet, J Campos, and T Dupuy. Visiting the bear den, a journey in the land of (cyber-) espionage. *RECON 2016, Montreal*, 2016.
8. José Cambronero, Sumit Gulwani, Vu Le, Daniel Perelman, Arjun Radhakrishna, Clint Simon, and Ashish Tiwari. Flashfill++: Scaling programming by example by cutting to the chase. *Proc. ACM Program. Lang.*, 7(POPL):952–981, 2023.
9. Christian Collberg and Jasvir Nagra. *Surreptitious Software: Obfuscation, Watermarking, and Tamperproofing for Software Protection*. 2009.
10. Christian Collberg, Clark Thomborson, and Douglas Low. A taxonomy of obfuscating transformations, 1997.
11. Yuantian Ding and Xiaokang Qiu. Enhanced enumeration techniques for syntax-guided synthesis of bit-vector manipulations. *Proc. ACM Program. Lang.*, 8(POPL):2129–2159, 2024.
12. Sumit Gulwani. Programming by examples. *Dependable Software Systems Engineering*, 45(137):3–15, 2016.
13. Grégoire Menguy, Sébastien Bardin, Richard Bonichon, and Cauim de Souza Lima. Search-based local black-box deobfuscation: Understand, improve and mitigate. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, CCS '21, 2021.
14. Saswat Padhi, Elizabeth Polgreen, Mukund Raghothaman, Andrew Reynolds, and Abhishek Udupa. The sygus language standard version 2.1. *CoRR*, abs/2312.06001, 2021.
15. Saswat Padhi, Rahul Sharma, and Todd D. Millstein. Data-driven precondition inference with learned features. In Chandra Krintz and Emery D. Berger, editors, *Proceedings of the 37th ACM SIGPLAN Conference on Programming Language Design and Implementation, PLDI 2016*, 2016.
16. Jonathan Salwan, Sébastien Bardin, and Marie-Laure Potet. Symbolic deobfuscation: from virtualized code back to the original. In *5th Conference on Detection of Intrusions and malware & Vulnerability Assessment (DIMVA)*, 2018.

17. Monirul Sharif, Andrea Lanzi, Jonathon Giffin, and Wenke Lee. Automatic reverse engineering of malware emulators. In *2009 30th IEEE Symposium on Security and Privacy*, 2009.
18. Jon Stephens, Babak Yadegari, Christian S. Collberg, Saumya Debray, and Carlos Scheidegger. Probabilistic obfuscation through covert channels. In *2018 IEEE European Symposium on Security and Privacy, EuroS&P 2018*, 2018.
19. VM Protect Software. VMProtect Software Protection. http://vmpsoft.com, 2020.
20. Yongxin Zhou, Alec Main, Yuan X. Gu, and Harold Johnson. Information hiding in software with mixed boolean-arithmetic transforms. In *Proceedings of the 8th International Conference on Information Security Applications*, WISA'07. Springer-Verlag, 2007.