Code Obfuscation

Why is This Still a Thing?

Christian Collberg
Department of Computer Science
University of Arizona
collberg@gmail.com
<http://collberg.cs.arizona.edu>

ABSTRACT

Early developments in code obfuscation were chiefly motivated by the needs of Digital Rights Management (DRM) [7]. Other suggested applications included intellectual property protection of software [4] and code diversification to combat the monoculture problem of operating systems [2].

Code obfuscation is typically employed in security scenarios where an adversary is in complete control over a device and the software it contains and can tamper with it at will. We call such situations the *Man-At-The-End* (MATE) [3] scenario. MATE scenarios are the best of all worlds for attackers and, consequently, the worst of all worlds for defenders: Not only do attackers have physical access to a device and can reverse engineer and tamper with it at their leisure, they often have unbounded resources (time, computational power, etc.) to do so. Defenders, on the other hand, are often severely constrained in the types of protective techniques available to them and the amount of overhead they can tolerate.

In other words, there is an asymmetry between the constraints of attackers and defenders. Moreover,

- DRM is becoming less prevalent (songs for sale on the Apple iTunes Store are no longer protected by DRM, for example);
- there are new cryptographically-based obfuscation techniques [1] that promise provably secure obfuscation;
- secure enclaves [5] are making it into commodity hardware, providing a safe haven for security sensitive code; and
- recent advances in program analysis [12] and generic de-obfuscation [13] provide algorithms that render current code obfuscation techniques impotent.

Thus, one may reasonably ask the question: “Is Code Obfuscation Still a Thing?”

Somewhat surprisingly, it appears that the answer is yes. In a recent report, Gartner [14] lists 19 companies active in this space (8 of which were founded since 2010) and there are still (in 2017) many papers published on code obfuscation, code de-obfuscation, anti-tamper protection, reverse engineering, and related technologies.

One of the reasons for this resurgence of code obfuscation as a protective technology is that, more and more, we are faced with applications where security-sensitive code needs to run on unsecured

endpoints. In this talk we will show MATE attacks that appear in many novel and unlikely scenarios, including smart cars [6], smart meters [9], mobile applications such as *Snapchat* and smartphone games, Internet of Things applications [8], and ad blockers in web browsers [11]. We will furthermore show novel code obfuscation techniques that increase the workload of attackers [10] and which, at least for a time, purport to restore the symmetry between attackers and defenders.

CCS CONCEPTS

• **Security and privacy** → **Digital rights management; Software reverse engineering; Malware and its mitigation; Hardware attacks and countermeasures;**

KEYWORDS

Software protection, code obfuscation, reverse engineering, Man-At-The-End, diversification, digital rights management, software watermarking, anti-tamper

BIOGRAPHY

Christian Collberg is a Professor in the Department of Computer Science at the University of Arizona. Prior to arriving in Tucson he worked at the University of Auckland, New Zealand, and before that got his Ph.D. from Lund University, Sweden. He has also held a visiting position at the Chinese Academy of Sciences in Beijing, China, and taught at universities in Russia and Belarus.

Dr. Collberg’s main research interest is Software Protection, the use of code obfuscation, software watermarking, anti-tamper, and related techniques to protect software that runs on endpoints under the control of an adversary. He is the co-author of the first book on software protection, *Surreptitious Software: Obfuscation, Watermarking, and Tamperproofing for Software Protection*, published in Addison-Wesley’s computer security series. It has also been translated into Portuguese and Chinese.

In addition to his security research, Dr. Collberg is an advocate for Reproducibility, Repeatability, and Sharing in Computer Science. He maintains the site FindResearch.org which aims to be the most authoritative and complete catalog of research artifacts (i.e., code and data) related to Computer Science publications.

ACKNOWLEDGMENTS

This work was funded in part by the NSF under grants CNF-1145913 and CNF-1145913.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

CODASPY '18, March 19–21, 2018, Tempe, AZ, USA

© 2018 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-5632-9/18/03.

<https://doi.org/10.1145/3176258.3176342>

REFERENCES

- [1] Boaz Barak. 2016. Hopes, Fears, and Software Obfuscation. *Commun. ACM* 59, 3 (Feb. 2016), 88–96. <https://doi.org/10.1145/2757276>
- [2] Frederick B. Cohen. 1993. Operating system protection through program evolution. *Computers & Security* 12, 6 (1993), 565–584.
- [3] Christian Collberg and Jasvir Nagra. 2009. *Surreptitious Software: Obfuscation, Watermarking, and Tamperproofing for Software Protection*. Addison-Wesley, 816 pages.
- [4] Christian S. Collberg and Clark D. Thomborson. 2002. Watermarking, Tamper-Proofing, and Obfuscation-Tools for Software Protection. *IEEE Trans. Software Eng.* 28, 8 (2002), 735–746.
- [5] Victor Costan and Srinivas Devadas. 2016. Intel SGX Explained. *IACR Cryptology ePrint Archive* (2016), 86.
- [6] European Union Agency for Network and Information Security. 2016. Cyber Security and Resilience of Smart Cars — Good practices and recommendations. (Dec. 2016).
- [7] Karl L. Ginter, Victor H. Shear, W. Olin Sibert, Francis J. Spahn, and David M. Van Wie. 1996. US Patent 5,892,900: Systems and methods for secure transaction management and electronic rights protection. (1996).
- [8] Shohreh Hosseinzadeh, Sami Hyrynsalmi, and Ville Leppänen. 2016. *Obfuscation and Diversification for Securing the Internet of Things (IoT)*. Elsevier, 259–274.
- [9] Parks R.C. 2007. Sandia Report: Advanced Metering Infrastructure Security Considerations. (Nov. 2007).
- [10] Jon Stephens, Babak Yadegari, Christian Collberg, Saumya Debray, and Carlos Scheidegger. 2018. Probabilistic Obfuscation through Covert Channels. In *3rd IEEE European Symposium on Security and Privacy*.
- [11] Grant Storey, Dillon Reisman, Jonathan Mayer, and Arvind Narayanan. 2017. The Future of Ad Blocking: An Analytical Framework and New Techniques. *CoRR* abs/1705.08568 (2017).
- [12] Fish Wang and Yan Shoshitaishvili. 2017. Angr - The Next Generation of Binary Analysis. In *SecDev. IEEE*, 8–9.
- [13] Babak Yadegari, Brian Johannesmeyer, Ben Whitely, and Saumya Debray. 2015. A Generic Approach to Automatic Deobfuscation of Executable Code. In *IEEE Symposium on Security and Privacy*. IEEE Computer Society, 674–691.
- [14] Dionisio Zumerle and Manjunath Bhat. 2017. Market Guide for Application Shielding. (2017).